우리식조작체계문고

Linux 지령어편람

교육성교육정보쎈터

주체97(2008)

이 책을 내면서

위대한 령도자 **김정일**동지께서는 다음과 같이 지적하시였다.

《정보산업에 대한 인식을 바로 가지고 정보산업시대의 요구에 맞게 일하여야 합니다.》 (《**김정일**선집》 제15권, 195페지)

위대한 령도자 **김정일**동지께서 지적하신바와 같이 정보산업에 대한 인식을 바로 가지고 정보산업시대의 요구에 맞게 일하는것은 정보산업시대에 들어선 오늘 나라의 과학기술을 세계적수준으로 끌어올리며 인민경제의 정보화를 실현하는데서 매우 중요한 문제로 나선다.

정보기술을 소유하는데서 우리는 무엇보다도 먼저 콤퓨터의 가동기반인 조작체계에 대하여 잘 알아야 한다.

조작체계 Linux는 1991년에 처음으로 개발된 이후 세계의 수많은 프로그람작성자들이 이 개발사업에 참가하여 많은 변화와 발전을 이룩하였다. Linux는 특히 안전성과 보안의 측면에서 그 어느 조작체계에 비할바없이 우월하며 강력한 망기능을 지원하고있으므로 Internet상에서 가는곳마다 사용자들의 환영을 받고있다. 이제는 사용자들이 Linux의 설치에서부터 사용에 이르기까지 Windows대면부와 거의 다를바 없는 GUI환경에서 편리하게 작업할수 있게 되였다. 다시말하여 마우스를 한번 누르는것만으로도 원하는 결과를 얻을수 있게 되였다.

그러나 현재까지도 변하지 않고 사용자들이 즐겨 사용하는것이 있는데 그것이 바로 입력재촉문에서 건반으로 입력하고 실행하는 기초지령들이다.

Linux를 얼마나 능숙하게 다루는가 하는것은 주어진 조건에 알맞는 지령들을 빨리 찾아내고 어떻게 효과적으로 사용하는가를 보면 알수 있다. 《눈에는 익고 손에는 설다.》는 속담에 있는것처럼 지령의 기능을 알고있다고 해서 그것을 정통했다고 할수는 없는것이다. 임의의 조건에서도 지령을 효과적으로 사용하자면 그 지령 하나하나의 쓰임을 정확히 리해하는것이 중요하다.

특히 오늘 Linux체계에 기초한 우리 식 조작체계의 개발과 연구가 급속히 추진되고있는 조건에서 그에 익숙하자면 그 기초로 되는 Linux지령들에 대하여 잘 알아야 한다.

이러한 현실적요구로부터 우리는 우리 식 조작체계를 리용하게 되는 사용자들이 Linux에 보다 빨리 익숙해질수 있도록 Linux지령들을 편람형식으로 묶어 출판한다.

이 책에서는 독자들이 콤퓨터에 마주앉지 않고서도 아무러한 불편없이 학습할수 있게 지령경로와 형식, 추가선택들에 대한 설명을 주었다. 그리고 단순히 편람으로서가 아니라 독자들이 Linux체계를 다루는데서 제기될수 있는 여러가지 경우들을 고려하여 풍부한 실례와 함께 실행화면들을 주었으며 련관이 있는 지령들도 소개하였다. 뿐만아니라 다양한 개념과 함께 상식들에 대하여서도 구체적으로 설명함으로써 독자들이 체계적이면서도 명백한 지식을 가지도록 하였다.

다른것과 마찬가지로 누가 더 많이 다루어보고 경험을 쌓는가 하는데 Linux전문가로 될수 있는 열쇠가 있다. 우리는 나라의 정보과학기술을 세계최첨단수준으로 끌어올리겠다는 비상한 각오와 불타는 정열을 가지고 높은 프로그람기술을 소유함으로써 강성대국건설에 적극 이바지하여야 한다.

우리는 이 책이 Linux를 열망하는 독자들의 좋은 방조자가 되기를 기대하면서 기탄없는 의견을 보내주기 바란다.

교육성 교육정보쎈터로부터

차 례

alias

내부지령

지령을 간단히 사용하기

지령형식: alias name[=value]

복잡한 지령과 추가선택을 간단히 입력할수 있는 문자렬로 바꾼다.

추가선택을 포함한 긴 지령을 사용할 때마다 매번 입력하지 않고도 대신 alias지령으로 치환한 간단한 문자렬을 사용할수 있다. 실례로 rm지령을 리용하여 파일을 삭제할 때 파일을 지우겠는가고 다시 묻는 추가선택을 사용하지 않고서도 rm지령이 이 추가선택을 사용하고있으며 ls지령으로 파일목록을 보았을 때 색으로 구분되는것은 대부분의 환경에서 미리 alias로 해당 추가선택을 예약해놓았기때문이다.

쉘은 내부지령 alias와 unalias를 리용하여 이런 예약내용을 목록에 설정하기도 하고 해제하기도 한다. 즉 사용자가 어떤 지령을 입력하면 이 지령과 일치하는 alias가 있는가를 목록에서 확인하고 일치하는것이 있으면 원래의 지령과 바꾸어 실행한다.

쉘 프롬프트에서 alias를 입력해보자. 그러면 현재 체계에 정의된 alias의 목록을 볼수 있다. 이전에 alias련관지령을 실행해본적이 없다면 Linux를 설치할 때부터 기정으로 설정된것이다.

주의 alias에서는 메타문자를 사용할수 있으나 "="만은 사용할수 없다. 그것은 alias와 원래의 지령을 대응시킬 때 "="를 사용하기때문이다.

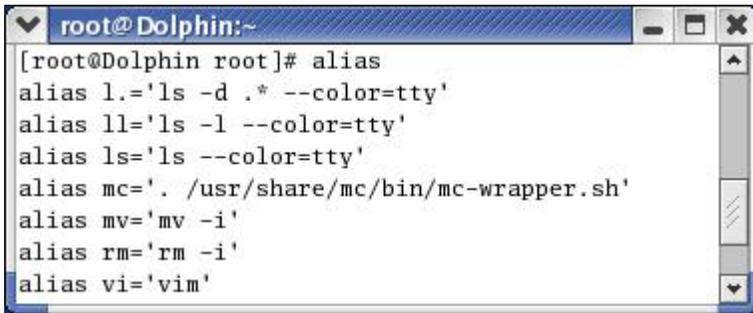
```
root@Dolphin:~
[root@Dolphin root]# alias
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mc='. /usr/share/mc/bin/mc-wrapper.sh'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim'
```

원래 cp지령은 복사할 파일이 이미 존재하는가에 관계없이 복사하지만 cp지령에 i추가선택을 붙이게 되면 같은 이름의 파일이 있는 경우 덧쓰기하겠는가를 물어본다. 이런

추가선택이 있으면 실수로 파일을 덧쓰는것과 같은 현상을 없앨수 있으므로 미리 cp지령에 i추가선택까지 주어서 cp alias를 지정하였다. 이와 같은 지령은 대개 기정으로 설정된 내용이다.

그러면 cp alias를 해제해보자. alias해제지령은 unalias이다.

```
# unalias cp
```

alias지령으로 예약된 내용을 보자.

cp alias가 없어졌다.

다시 cp alias를 생성하려면 다음과 같이 입력한다.

```
# alias cp='cp -i'
```

우와 같은 방법으로 자주 쓰는 지령과 추가선택을 간단하게 줄여 사용할수 있다. 실례로 자주 사용하는 tar지령과 추가선택을 다음과 같이 만들어놓으면 편리하다.

```
# alias tarx='tar xvpf'
# alias tarc='tar cvpf'
# alias tarz='tar xvpfz'
```

다른 지령도 같은 방법으로 응용할수 있다.

또한 alias를 설정한 상태에서 원래의 지령을 사용하려면 다음과 같은 방법들을 사용할수 있다.

```
# \cp 혹은 # /bin/cp
```

지령앞에 \(역사선)이 붙어있으면 alias를 무시하고 원래의 지령을 실행하라는 뜻이다. 또한 절대경로를 입력하여 지령을 실행할 경우에는 alias를 사용할수 없다.

련관지령

unalias alias를 해제한다.

apm

/usr/bin/apm

노트형콤퓨터의 전원관리

지령형식: apm[추가선택]

-V, --version: 판본정보를 보여준다.
-v, --verbose: APM BIOS의 판본과 전원상태정보를 보여준다.
-m, --minutes: 축전지의 남은 시간을 보여준다.(단위 : min)
-s, --sespend: 체계를 정지상태로 만든다.
-S, --standby: 체계를 대기상태로 만든다.
-i: AC전원을 사용할 때 APM의 체계정지, 대기지령을 무시한다. 노트형콤퓨터에서 축전지를 사용하지 않고 AC전원을 직접 사용하는 경우에 리용된다.
-n: AC전원을 사용할 때 APM의 체계정지, 대기지령을 무시하지 않도록 한다.

노트형콤퓨터와 같은 장치에서 축전지의 현재 남은 량을 확인하고 상태에 따라 체계를 대기/완료시키는 지령으로서 노트형콤퓨터에서 Linux를 사용하는 사람에게는 꼭 필요한 지령이다.

노트형콤퓨터에서 축전지의 관리는 아주 중요한 문제이다. apm지령은 APM (Advanced Power Management) BIOS의 정보가 들어있는 /proc/apm을 읽어들여 체계의 전원상태를 보여준다.

주의 JDK란 Sun에서 제공하는 Java개발환경을 말한다. 여기에는 Java콤파일러와 Java가상기계, 오유수정프로그람 등 응용프로그람개발을 위한 도구들이 포함되여있다.

그러면 노트형콤퓨터에서 축전지의 남은 시간을 확인해보자.

여기에서 5는 현재 상태에서 콤퓨터를 계속 리용하는 경우 앞으로 축전지를 사용할수 있는 시간을 min으로 환산한것이다.

탁상형콤퓨터나 혹은 노트형콤퓨터를 직접 전원에 련결한 상태에서 이 지령을 실행하면 다음과 같은 통보문이 나타난다.

련관지령

apmd　　전원관리데몬

baitstat_applet　　축전지잔량을 실시간으로 확인할수 있는 소응용프로그람

appletviewer　　/usr/java/bin/appletviewer

Java애플레드 실행

지령형식: /usr/java/bin/appletviewer [추가선택] URL

URL : 실행할 Java애플레트가 있는 주소 또는 경로
-debug : Java debuger(jdb)에서 애플레트를 실행한다.
-encoding name : 입력되는 HTML파일에 대한 부호화방식을 지정한다.
-J opt : Java지령에 사용하는 추가선택을 련결한다.

이 지령은 Internet상의 Java애플레트나 콤퓨터에 있는 Java애플레트를 따로 보기 위한것으로서 주로 애플레트가 정상적으로 동작하는가를 확인하고 그에 대한 오유를 수정하는데 사용된다. 애플레트를 콤파일하여 확인할 때마다 웨브열람기를 사용하는것보다 이 지령을 사용하는것이 더 효과적이다.

원래는 Solaris에서 사용하던 지령인데 Linux에서도 JDK를 설치하면 사용할수 있다. 애플레트 그자체는 불러들일수 없으며 애플레트가 포함된 HTML페지를 읽어 수행한다.

애플레트는 단독으로 실행할수 없다. 사용할 때에는 HTML파일안에 다음과 같은 내용을 삽입하여 불러들여야 한다.

```
<object width="pixelWidth" height="pixelHeight">
```

```
    <param name="code" value="MyClass class">
    <param name="object" value="serializedObjectOrJavaBean">
    <param name="codebase" value="classFileDirectory">
    ...
    (기타 필요한 내용)
</object>
```

직접 만든 애플레트인 MyClass를 시험해보고싶다면 먼저 우와 같은 내용을 MyClass.html이라는 이름으로 보관한다. 다음 이 MyClass.html을 불러들여 애플레트를 확인한다.

```
# appletviewer MyCalss.html
```

Java애플레트가 있는 웨브페지에 대한 오유수정은 불러들였을 때 웨브페지주소와 --debug추가선택을 함께 사용하여 진행한다.

```
# appletviewer MyClass.html
# appletviewer -debug http://www.sec.go.kp
```

주의 JDK를 설치했는데도 appletveiwer지령을 사용할수 없다면 환경변수 PATH를 확인하여야 한다.

애플레트(Applet) 용어

상식

애플레트는 소규모의 동화상이나 간단한 기능을 처리하는 작은 프로그람으로서 주로 Java언어와 같은 객체지향언어로 작성된다. 독립적으로가 아니라 다른 프로그람에 덧붙어 실행된다. 애플레트가 사용되는 대표적인것으로는 웨브페지에서 그림그리기를 할수 있는것이 있으며 우리가 사용하는 KDE나 Gnome환경에서 사용하는 간단한 기능의 프로그람도 애플레트라고 한다.

WWW를 사용하기전에 Windows에서 제공되던 소규모프로그람들, 실례로 Notepad나 Paint 등을《애플레트》라고 부르기도 하였지만 지금은 그런 표현을 거의 사용하지 않는다.

apropos

/usr/java/apropos

련관지령찾기

지령형식: apropos [열쇠어]

열쇠어: 검색하고싶은 문자렬

apropos는 whatis DB를 검색하여 지령과 그 지령에 대한 간단한 설명을 보여준다.
whatis는 whatis DB에서 해당 지령만 검색하여 출력하지만 apropos는 지령과 설명을 모두 검색하여 출력한다.

실례로 이런 상태를 생각해보자. Linux에서는 GIMP를 사용하지 않고서도 *.bmp 파일 등을 *.jpeg로 압축할수 있다. 그런데 그 지령이 생각나지 않을 경우에는 다음과 같이 입력하면 된다.

주의 GIMP는 GNU Image Manipulation Program의 략자로서 UNIX용 2차원도형작성프로그람이다.

```
[root@Dolphin root]# apropos jpeg
cjpeg      (1)- compress an image file to a JPEG file
djpeg      (1)- decompress a JPEG file to an image file
jpegtopnm  (1)- convert JPEG/JFIF file to portable pixmap or graymap
jpegtran   (1)- lossless transformation of JPEG files
pnmtojpeg  (1)- convert PNM image to a JFIF (JPEG) image
rdjpgcom   (1)- display text comments from a JPEG file
wrjpgcom   (1)- insert text comments into a JPEG file
```

이 지령을 사용하면 jpeg라는 단어가 들어가는 지령들을 다 찾아볼수 있다. 일반 화상파일을 JPEG파일로 만들어주는 지령은 맨 우에 있는 cjpeg이다. 또한 JPEG파일을 bitmap파일이나 그밖의 파일로 만들어주는 지령이 djpeg라는것도 확인할수 있다.

련관지령

man	-k추가선택을 사용하면 apropos와 같은 기능을 한다.
whatis	whatis DB에서 지령과 일치하는 내용을 검색하여 보여준다.

whatis DB

지령과 그에 대한 설명을 보관한 파일이다. /var/cache/man/에 whatis라는 행으로 구분되여있는 본문 DB파일로 존재한다. appropos와 whatis지령은 이 파일에서 해당한 지령을 검색하여 그 내용을 보여준다. whatis DB는 makewhatis 지령으로 파일을 생성하거나 갱신할수 있다.

whatis DB의 갱신

makewhatis는 체계에 등록된 최신지도서페지를 검색하여 whatis DB를 갱신한다.

```
# /usr/sbin/makewhatis u
```

arch

/bin/arch

CPU의 류형보기

지령형식: arch

체계의 CPU류형을 보여준다. X86의 Intel계렬 CPU는 i386, i486, i586, i686, athlon으로 보여주며 그밖에 CPU는 alpha, sparc, arm, m68k, mips, ppc와 같이 보여준다.

인텔계렬 CPU

용어

상식

여러 회사들에서 다양한 종류의 CPU들을 만들어내고있지만 그중에서 널리 사용되고있는 CPU는 Intel계렬 CPU이다.

다음의 표는 Intel계렬 CPU의 발전력사를 보여준다.

4004(1971년)	Intel이 처음으로 만든 4bit CPU이며 기본적인 산수계산만 가능.
8008(1972년)	4004의 8bit판본. 0.2MHz로 동작.
8080(1974년)	첫번째 OS인 CP/M을 사용하는 Z80 CPU의 모형.
8086(1978년)	최초의 16bit자료모선을 사용. 4.77, 8, 10MHz속도로 동작.
8088(1979년)	8086과 같은 16bit등록기를 사용하지만 70년대 후반기 하드웨어와 호환성을 유지하기 위해 외부모선을 8bit로 다시 설계한 제품. 4.77, 8MHz로 동작.
80186/80188 (1979년)	8086과 8088을 기반으로 내부마이크로코드를 개선하고 CPU의 모양을 PGA의 형태로 만든 제품.
80286 (1979년)	16bit자료모선에 24bit주소화와 16bit등록기를 사용. 6MHz로 동작하지만 4.77MHz인 8088 CPU의 4배되는 성능을 낼수 있음.
80386(DX/SX/SLX 1985년)	최초의 32bit Intel계렬의 CPU. 실방식, 보호방식외에 가상 8086방식(여러개의 가상 8086방식을 모방할수 있음)을 적용하여 처음으로 다중과제를 처리할수 있게 되였다.
80486(DX/SX/DX2/DX4) (1989년)	80386에 L1 cache내장, clock multiple, 3.3V지원 등을 추가하였으며 현재의 Pentium CPU 및 이후 CPU들의 설계모형이 되였다.

i386

rpm파일 등을 설치하는 과정에 i386이라는 말을 자주 보게 된다. 이것은 이 rpm파일이 80386과 아래준위호환성을 가지는 대부분의 콤퓨터에서 원활하게 설치될수 있게 80386환경에서 만들어졌다는 의미이다.(i686 등으로 하려면 추가선택을 주어야 한다.)

arp

/sbin/arp

arp완충기 보기

 지령형식: arp [추가선택]

-v : arp상태를 보여준다.
-t type : ARP캐쉬에 등록되여있는 류형을 검색한다. ether(Ethernet) 또는 ax25(Ax25 packet radio) 등이 있으며 기정은 ether이다.
-a [hosts] : 등록된 콤퓨터중에서 지정한 콤퓨터의 내용을 보여준다. 콤퓨터를 지정하지 않으면 등록된 콤퓨터들을 모두 보여준다.
-d [host] : 지정한 콤퓨터를 목록에서 삭제한다.
-shost hardware-address : 콤퓨터의 하드웨어주소 즉 콤퓨터의 MAC주소를 추가한다. Ethernet기판인 경우 6자리의 16진수로 되여있다.
-f file : 파일에 있는 목록을 추가한다.

련결하려고 하는 체계의 MAC(Medium Access Control)주소를 확인하는 TCP/IP지령이다. 일반적으로 체계들사이의 통신에서는 상대방의 MAC주소가 필요하다. 이때 arp는 ARP(Address Resolution Protocol)를 리용하여 련결하려고 하는 체계의 IP에 신호를 보내여 MAC주소를 받아온다.

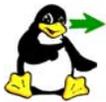
Subnet의 arp정보는 련결효률을 높이기 위해 /proc/net/arp에 저장된다. 완충기에 저장된 정보는 추가 또는 삭제할수 있다. 저장된 arp완충기의 내용을 구체적으로 보자면 다음과 같이 실행한다.

```
root@Dolphin:~
[root@Dolphin root]# arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.8.84             ether   00:0C:76:00:DD:45   C                     eth0
192.168.8.81             ether   00:0C:76:00:DC:C7   C                     eth0
192.168.8.89                     (incomplete)                              eth0
Entries: 3      Skipped: 0      Found: 3
```

hana라는 콤퓨터에 대한 정보만 알고싶다면 다음과 같이 한다.

```
root@Dolphin:~
[root@Dolphin root]# arp -a hana
hana (192.168.8.84) at 00:0C:76:00:DD:45 [ether] on eth0
```

경우에 따라 체계에 기존 Ethernet기판대신 새 Ethernet기판을 설치했을 때 내부망련결이 잘 되지 않는 경우가 있다. 이것은 arp캐쉬가 기존 Ethernet기판의 MAC주소를 저장하고있으므로 같은 IP를 사용하더라도 인식하지 못하기때문이다. 이 경우에는 arp -d지령을 사용하여 정보를 삭제한 다음 다시 해본다.

```
# arp -d hana: 콤퓨터이름이 hana인 정보삭제
```

련관지령

arping	대상주소에 ARP파케트를 보낸다.
arpwatch	Ethernet/IP주소의 경로를 추적한다.
arpsnmp	Ethernet/IP주소의 경로를 추적한다.
tcpdump	망대면부에서의 파케트머리부들을 출력한다.

ARP

Address Resolution Protocol의 략자로서 망상에서 IP주소를 MAC주소와 대응시키기 위해 사용하는 규약이다. 실례로 콤퓨터 A가 콤퓨터 B에 IP파케트를 전송하려고 할 때 콤퓨터 B의 MAC주소를 모르는 경우 ARP규약을 사용하여 목적지 B의 IP주소와 송신MAC주소 FFFFFFFFFFFF를 가지는 ARP파케트를 망상에 전송한다. 콤퓨터 B는 자기의 IP주소가 있는 ARP파케트를 수신하면 자기의 MAC주소를 A에게 돌려보낸다. 이와 같은 방식으로 수집된 IP주소와 이에 해당하는 MAC주소정보는 매 IP콤퓨터의 ARP캐쉬라고 하는 기억기에 표형태로 저장되여 다음 파케트를 전송할 때 다시 사용할수 있다. ARP와는 반대로 IP콤퓨터가 자기의 물리망주소는 알지만 IP주소를 모르는 경우 봉사기에 IP주소를 요청할 때에는 RARF(Reversek ORP)를 사용한다.

규약 (Protocol)

규약은 주콤퓨터들사이 또는 주콤퓨터와 말단콤퓨터사이에 정보를 교환하는 통신규약이다. 통신규약이란 련결된 콤퓨터들사이의 접속이나 절단, 통신방식, 주고받을 자료의 형식, 오유검출방식, 코드변환방식, 전송속도 등에 대하여 규정한것이다.

기종이 다른 콤퓨터들이 서로 통신을 하자면 표준규약을 리용하여 통신망을 구축해야 한다. 이와 같은 표준규약중에서 가장 대표적인것이 Internet에서 널리 사용하고있는 TCP/IP이다.

MAC주소 (MAC address)

망기판을 제작하는 회사들에서 기판마다에 붙힌 48bit의 고유번호이다. 망이 IP주소만으로 운영되다가 문제가 생기는 경우에 체계를 구분할수 있는 정보가 필요하였다. 간단한 실례로 IP충돌이 일어났을 경우 그것을 알수 있는 다른 정보가 필요한것이다. 그래서 생성된것이 MAC주소이다. 두대의 콤퓨터의 IP주소를 같게 하여 문제가 생긴 경우에도 매 체계는 이 망기판의 고유번호로 구분할수 있다.

arping

/sbin/arping

망련결확인

지령형식: arping [추가선택]

-c count : arping의 개수를 지정한다.
-w timeout : w추가선택 다음에 시간(s)을 지정하여 그 시간(s)까지만 apring을 실행한다.
-I device : Ethernet장치를 지정한다.
-s source : 원천(source) IP를 지정한다.

ARP요청을 리용한 망련결확인지령이다.

ping의 변형으로서 ICMP파케트대신 ARP요청과 응답으로 사용중에 있는 IP주소에 해당한 MAC주소를 찾을수도 있고 MAC주소가 있을 때 해당한 IP주소를 찾을수도 있다.

또한 ICMP파케트를 차단한 콤퓨터라도 련결성만 검사하고 싶을 때 효과적으로 리용할수 있다.

다음의 실례는 192.168.8.57이라는 IP주소를 가지고있는 콤퓨터에 arping을 세번 보내여 확인하는 실례이다.

체계에 망장치가 여러개 장비되여있을 때 매 장치들사이의 련결성을 검사하고싶다면 "-I"추가선택을 리용하여 장치를 지정하여준다.

련관지령

ping 망련결성을 분석한다.

UDP (User Datagram Protocol)

UDP는 IP를 사용하는 망에서 콤퓨터들사이에 통보문이 교환될 때 제한된 봉사만을 제공하는 통신규약이다. TCP와는 달리 도착하는 전송과정에 자료파케트들의 순서를 보장하지 않는다. 대신 처리속도가 빠르므로 교환해야 할 자료가 매우 적은 경우 TCP보다 유용하게 사용할수 있다.

TCP (Transmission Control Protocol)

TCP는 Internet상의 콤퓨터들사이에서 자료를 통보문형태로 보내기 위해 IP와 함께 사용되는 규약으로서 OSI통신모형에서는 4번째 층인 전송층에 속한다.

TCP는 말단응용프로그람들사이에 통보문들이 교환되는 동안에도 련결이 유지되기때문에 련결지향규약으로도 알려져있으며 전송과정에 자료파케트들의 순서를 보장한다.

ICMP (Internet Control Message Protocol)

ICMP는 콤퓨터봉사기와 Internet관문에서 통보문을 조종하고 오유를 알려주는 규약이다. 경로기가 망경로를 설정할수 없거나 자료를 전달할수 없는것과 같은 망상의 문제를 해결하면서 정보를 전송하는 규약이다. ping지령도 이 규약을 사용하게 된다.

at

/usr/bin/at

작업예약

지령형식: at [추가선택] 시간 [날자] [+증가시간]

시간 : 지령이나 스크립트를 실행할 시간을 지정한다.
+증가시간 : 증가시간만큼 지난 후 실행한다.
-q queue : queue이름을 지정한다. 대기렬이름으로는 a~z, A~Z가 쓰일수 있다. 대기렬이름을 지정해주지 않으면 at는 a를 batch는 b를 사용한다. 자모순으로 뒤에 놓이는 이름의 대기렬은 더 큰 대기시간을 가진다. =대기렬은 현재 수행되는 작업의 대기렬로 예약되여있다. 이름이 대문자로 된 대기렬에 추가된 작업은 batch의 작업처럼 처리된다.
-m : 출력결과가 없더라도 작업이 완료될 때 사용자에게 우편을 보낸다.
-f 파일이름 : 스크립트파일 등을 실행해줄 때 사용한다.
-l : 예약된 작업목록을 보여준다.
-v : 작업이 수행될 시간을 보여준다.
-d : 예약된 작업을 삭제한다. (Linux)
-r: 예약된 작업을 삭제한다. (Unix)

at는 등록된 지령을 지정한 시간에 실행하게 한다. 즉 체계의 부하가 적은 시간에 실행시키거나 예약된 시간에 꼭 실행해야 하는 작업을 자동으로 처리하려고 할 때 사용한다.

주의 at지령을 실행하려면 atd데몬이 동작하고있어야 한다.

at에 의해 실행할 시간설정

예약작업기능이므로 시간에 대한 설정이 중요하다. at의 시간설정은 날자와 시간을 정해주는 시간설정과 현재시간 혹은 정해진 시간에서 증가시간으로 설정해주는 방법이 있다.

시간표현방법

분 류	형 식	설 명	실 례
시간	hh:mm	hh(시간), mm(분)으로 설정	16:18
	am, pm	표시가 없는 경우 Linux나 Unix에서는 하루를 24시간으로 표현한다. am과 pm을 리용하여 12시간 단위로 표현이 가능하다.	4:18 pm
	midnight noon teatime now	midnight(00:00), noon(12:00), teatime(16:00) 같은 형태로도 시간을 지정할수 있다. now는 현재시간이며 증가시간설정에 많이 쓰인다.	
날자	yyyy-mm-dd	일반적인 날자구성으로 표현	2002-4-13
	month num	4월 13일은 April 13으로 표현한다. 혹은 략자로 Apr 13으로 표현할수 있다.	April 13
today tomorrow		오늘 또는 래일로 지정할수 있다. 매 요일별로도 표현할수 있다.	

test1파일을 만들어주는 지령인 touch test1이라는 지령을 at에 등록하여 2005년 2월 28일 0시에 실행해보자. 2005년 2월 28일 0시는 00:00 2005-02-28 또는 midnight February 28 등으로 시간을 표현해줄수 있다. 지령을 다 입력한 다음에는 [Ctrl+D]건을 눌러 편집상태에서 탈퇴한다.

주의 한개의 지령뿐아니라 여러개의 지령을 함께 등록하여 사용할수도 있다.

예약된 작업 확인

at에 예약된 작업정보

작업번호	날자	시간	대기렬이름	사용자
1	2005-02-28	00:00	a	root

작업번호: at에 등록된 작업번호이다. 등록하는 순서대로 번호가 붙으며 at -d, 또는 atrm을 리용하여 at작업을 삭제할 때 사용한다.
날자/시간: 등록된 지령이 실행될 시간이다.
대기렬이름: -q추가선택을 리용하여 대기렬이름을 지정한다.
사용자: at에 작업을 예약한 사용자이다.

증가시간을 리용한 설정

지금으로부터 6시간후에 Control.sh스크립트를 실행하려면 다음과 같이 -f추가선택을 사용하여 예약작업으로 등록한다.
또한 "<"을 리용해도 같은 결과를 얻을수 있다.

```
# at now +6 hours -f Control.sh
```

혹은

```
# at now +6 hours < Control.sh
```

예약한 작업을 확인해보자.

1번에 예약된 작업을 삭제하고싶다면 다음과 같이 삭제할 작업번호를 지정한다.

at의 사용권한 설정

at.allow와 at.deny파일들을 리용하여 일반사용자의 at지령사용권한을 지정해줄수 있다.

/etc/at.allow파일이 있으면 at.allow에 기록된 사용자만 at지령을 사용할수 있으며 /etc/at.allow파일이 없으면 /etc/at.deny에 기록되지 않은 사용자만이 이 지령을 사용할수 있다. 두 파일이 모두 없다면 체계관리자(root)만 사용할수 있다.

련관지령

atq	예약된 작업목록을 보여준다.
atrm	예약된 작업목록을 삭제한다.
batch	체계의 부하상태에 따라 지령을 실행한다.
cron	정기적으로 예약된 작업을 수행한다.

at와 cron

at와 cron은 작업을 예약해서 실행한다. at는 일정한 시간을 정하기 쉽기때문에 비정기적인 시간을 예약하는데 사용하는것이 좋으며 정기적으로 수행해야 하는 작업인 경우에는 cron을 사용하는것이 더 편리하다.

cron은 월단위, 주단위, 일단위, 시간단위로 정기적인 시간설정을 할수 있다.

	At	Cron
공통점	작업을 예약하여 실행한다.	
쓰이는데	비정기적인 작업예약	정기적인 작업예약
관련데몬	Atd	crond
사용권한	/etc/at.allow 허가 /etc/cron.allow 허가	/etc/at.deny 거부 /etc/cron.deny 거부

UNIX시간 (Unix time)

date +%s지령을 사용하여 확인할수 있다. UNIX의 time()함수는 1970년 1월 1일 0시부터 현재까지 경과된 시간을 s단위로 정수값으로 반환하여 시간을 계산하고있다. time_t변수는 1970년 1원 1일 0시부터의 초계수값을 정수로 나타낸다. 이로부터 파일이름을 겹치지 않게 만들기 위해 UNIX시간을 리용하여 파일이름에 《시간도장》을 붙이는 경우가 많다.

문제점은 UNIX날자값이 4byte공간에 저장되여있기때문에 이 값을 32bit의 부호있는 옹근수로 표현하게 된다. 따라서 현재상태로는 2038년까지의 시간만 계산할수 있다.

Atq

/usr/bin/atq

예약작업보기

지령형식: atq [추가선택]

-q queue : 지정된 대기렬에 예약된 작업만을 보여준다.
-V : 판본정보를 보여준다.

at에 의해 예약된 작업목록을 보여준다. at -l과 같은 기능을 한다.

이중에서 일정한 대기렬에 예약된 작업을 보기 위해서는 atq -q라고 입력한다. 아래의 화면은 대기렬 t에 예약된 작업을 보여준다.

련관지령

at 작업을 일정한 시간에 예약한다.
atrm 예약된 작업목록을 삭제한다.

atrm

usr/bin/atrm

예약작업삭제

지령형식: atrm 작업번호

작업번호 : 현재 예약되여있는 작업번호

at를 사용하여 예약한 작업중에서 불필요한 내용을 삭제한다.
atq를 사용하여 예약된 작업목록을 확인한 다음 첫 마당에 있는 작업번호를 확인하자.

```
root@Dolphin:~
[root@Dolphin root]# atq
7       2005-02-28 12:00 t root
6       2005-02-28 18:00 a root
8       2005-02-28 00:00 a root
9       2005-02-28 05:28 a root
```

이 가운데서 2005년 2월 28일 12시에 예약된 작업을 삭제해보겠다. 이 작업의 작업번호는 7이므로 다음과 같이 삭제한다.

```
# atrm 7
```

정확히 삭제되였는가를 확인해보자.

```
root@Dolphin:~
[root@Dolphin root]# atq
6       2005-02-28 18:00 a root
8       2005-02-28 00:00 a root
9       2005-02-28 05:28 a root
```

련관지령

at	작업을 일정한 시간에 예약한다.
atq	예약된 작업목록을 보여준다.
batch	체계부하에 따라 지령을 실행한다.
cron	정기적으로 예약된 작업을 실행한다.

awk

/bin/awk

패턴처리언어

지령형식: awk [추가선택] -F 'script' [변수=값] [파일]

스크립트파일 : awk스크립트로 작성된 파일
변수 : awk내의 변수를 지정한다.
파일 : 대상파일
-F : 구분자를 나타낸다. -F로 구분자를 지정하지 않을 경우에는 공백을 구분자로 사용한다.
-f : 스크립트파일을 리용할 경우 사용한다.

표준입력으로 값을 받아 awk스크립트를 통하여 원하는 표준출력을 얻어낸다.

awk는 1977년에 AT&T연구소의 Alfred V. Aho, Peter J. weinverger, Brian W. Kernighan 세 사람이 만들었다. awk라는 이름도 이 세 사람의 이름의 첫글자들을 써서 만들었다. Linux에서는 1986년 Paul Rubin과 Jay Fenlason에 의해 GNU판본의 awk가 만들어졌다.

awk는 일정한 규칙을 가지고있는 자료를 처리하여 계산, 통계, 비교분석 혹은 려파기능을 통한 자료추출 등에 다양하게 사용될수 있다. sed와 비슷한 기능을 수행한다고 볼수 있다.

awk는 단순히 지령으로 사용될수도 있고 다른 스크립트에서 sed와 함께 리용될수도 있으며 awk만의 스크립트파일도 작성할수 있다.

지령사용

현재등록부의 파일정보를 보기 위해 ls -l지령을 리용하여 파일목록을 출력한다.

이 목록가운에서 파일의 호출권한과 파일이름만을 보려면 awk를 리용하여 쉽게 편집할수 있다. 다음의 실레는 awk가 ls -l지령으로 출력되는 자료를 흐름선(|)으로 받아 파일의 공백을 기준으로 첫번째 마당과 9번째 마당을 출력한다.

화면에서 보는바와 같이 awk로 려과하여 파일의 권한과 파일이름만을 출력한다. 우와 같이 awk는 지정하는 마당들을 구분자로 구분하고 print지령으로 출력하게 된다.

ls -l로 출력되는 행을 공백으로 구분하면 아래와 같다. 매개 마당은 공백을 기준으로 구분된다.($1~$9) $0은 모든 마당을 의미한다.

-rw-r--r--	2	root	root	23376	5	28	02:30	mbox
파일의 호출권한	하드련결번호	사용자	그룹	용량	월	일	시간	파일이름
$1	$2	$3	$4	$5	$6	$7	$8	$9

파일목록중에서 용량이 1MB인 파일과 그의 용량을 출력하려면 awk의 조건문을 사용할수 있다. 1MB=1048576bit이기때문에 용량을 나타내는 5번째 마당가운데서 1048576인 행을 출력하며 그중에서도 5번째와 9번째 마당만 출력한다.

```
root@Dolphin:~
[root@Dolphin root]# ls -l | awk '$5=1048576 {print $5,$9}'
1048576
1048576 anaconda-ks.cfg
1048576 install.log
1048576 install.log.syslog
1048576 root
```

awk를 리용하면 지정한 문자렬이 있는 행을 검색하여 출력할수 있다.

우의 화면은 /etc/group파일에서 root계정을 포함한 행만을 검색하여 출력하였다.

":"표시로 구분된 마당중에서 제일 먼저 나오는 그룹이름만 출력하려면 다음과 같이 할수 있다. 구분자는 기정으로 공백이지만 "-F:"로 지정할수도 있다.

```
root@Dolphin:~
[root@Dolphin root]# awk -F: /root/'{print $1}' /etc/group
root
bin
daemon
sys
adm
disk
wheel
```

/root/로 검색한 패턴부분에 정규표현식을 사용하면 보다 자세한 패턴을 검색할수 있다.

스크립트파일의 리용

시작(Begin)	스크립트의 시작을 알리는 단계이다.
실행(Routin)	스크립트를 수행하는 단계이다.
끝(End)	마무리단계이다. 결과를 출력한다.

awk의 스크립트구조는 시작, 실행, 끝 이렇게 3단계로 나누어져있다.

awk는 여러 연산자나 순환지령사용법 등이 C언어와 같으므로 C를 알고있는 사용자는 쉽게 사용할수 있다.

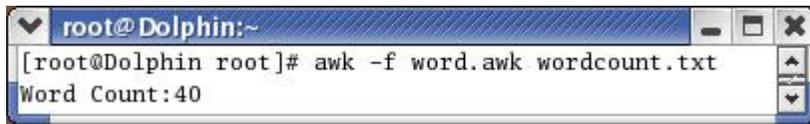
다음의 스크립트는 파일의 단어수를 검사하는 실례이다. wc라는 지령으로 쉽게 단어개수를 알아낼수도 있지만 여기서는 간단한 awk스크립트실례로 awk스크립트사용법을 알아보자.

awk스크립트	구 조	설 명
#/bin/awk	선언	awk로 스크립트가 실행될수 있게 선언한다.

BEGIN{
word=0;
}
	시작(Begin)	변수 word를 0으로 초기화한다.

{
word+=NF;
}
	실행(Routin)	NF는 매 행의 마당수를 나타내는 awk체계변수이다. awk는 구분자가 정의되여있지 않을 때에는 공백을 구분자로 사용하므로 매 행의 단어수가 NF로 들어가고 "+="연산자에 의해서 마지막 행의 단어개수까지 더해준다.

END {
print 'Word Count:
' word;
}
	끝(End)	결과를 출력한다.

작성된 스크립트는 -f추가선택으로 실행한다. wordcount.txt라는 임의의 본문문서를 방금 작성한 스크립트로 검사해보자. 작성된 awk스크립트파일을 실행하려면 awk -f [스크립트 파일이름]형식으로 사용한다.

awk체계변수

변 수	내 용
$0	모든 입력행
$n	입력행에서 n번째 마당값
ARGC	지령행인수의 개수를 포함하는 변수
ARGV	지령행인수를 포함하는 배렬
ENVRON	환경변수들을 모아둔 관계형배렬
FILENAME	현재파일이름
FS	구분기호를 정의하는 변수(기정으로 공백을 사용)
FNR	입력파일의 전체 행수
NF	현재행의 마당수

NR	현재행의 번호
OFMT	수자에 대한 출력형식
OFS	출력되는 마당을 구분하는 변수(기정으로는 빈 행)
ORS	출력되는 행을 구분(기정으로는 newline을 사용)
RLENGTH	지정한 패턴으로 검색되여나온 문자렬의 길이
RS	입력되는 행을 구분(기정으로는 newline을 사용)
RSTART	지정한 패턴으로 검색되여나온 문자렬의 가장 앞부분

awk 연산자

C를 참조하여 만들어졌으므로 C언어의 연산자와 사용법이나 종류가 거의 같다.

연산자	설　　명
?	조건연산. 사용자로 등록된 ID가 user1, user2, user3으로 되여있고 그중 검색하고싶은 내용이 1/2/3중에서 어느것인지 명확하지 않을 경우 "?"를 리용하여 모두 검색할수 있다. # awk /user?/ /etc/passwd user1:x:516:516::/home/user1:/bin/bash user2:x:517:517::/home/user2:/bin/bash user3:x:518:518::/home/user3:/bin/bash
||, &&, !	론리연산. 각각 론리더하기, 론리곱하기, 론리부정
~, !~	검색된 패턴에 부합되는것을 참으로 사용하려면 "~", 거짓으로 사용하려면 "!~"를 사용한다.
<,<=,>,>=, !=,==	비교연산자
+,-,*,/,%,^	더하기, 덜기, 곱하기, 나누기, 나머지, 제곱
++, --	증가연산자, 감소연산자

a2p

/usr/bin/a2p

awk-perl변환

지령형식: a2p [추가선택] 파일이름

-F<문자> : 구분자를 정의한다. awk의 마당을 나누는 기준이 되는 구분자변수인 FS변수를 지정된 문자로 정의한다.

-o : 이전 awk형식의 스크립트를 변환한다.

awk스크립트를 perl스크립트로 변환하는 지령이다.

awk지령에서 실례로 만든 간단한 단어계수스크립트를 변환해보자. word.txt라는 문서의 단어수를 검사해보자.

이 awk스크립트를 perl스크립트로 변환해보자.

```
root@Dolphin:~
[root@Dolphin root]# a2p word.awk
#!/usr/bin/perl
eval 'exec /usr/bin/perl -S $0 ${1+"$@"}'
    if $running_under_some_shell;
                        # this emulates #! processing on NIH machines.
                        # (remove #! line above if indigestible)

eval '$'.$1.'$2;' while $ARGV[0] =~ /^([A-Za-z_0-9]+=)(.*)/ && shift;
                        # process any FOO=bar switches

$[ = 1;                 # set array base to 1
$, = ' ';               # set output field separator
$\ = "\n";              # set output record separator

$word = 0;

while (<>) {
    chomp;      # strip record separator
    @Fld = split(' ', $_, 9999);

    $word += $#Fld;
}

print 'Word Count:' . $word;
```

a2p는 perl스크립트로 변환하고 표준출력으로 변환된 문서를 화면에 보여준다. 이 표준출력을 word.perl이라는 perl스크립트파일로 만들어 실행하면 같은 결과를 얻을수 있다.

주의 표준출력에 대한 내용은 awk를 참고하자.

s2p sed스크립트를 perl스크립트로 변환

badblocks

/sbin/badblocks

불량블로크의 검사

지령형식: badblocks [추가선택] [장치] [블로크수]

장치 : 검사할 장치를 지정한다.
블로크수 : 검사할 장치의 블로크개수
-b 블로크크기 : 블로크크기를 byte수로 나타낸다.
-o 파일 : 불량블로크 검사내용을 파일에 보관한다.
-v : 불량블로크 검사내용을 상세히 보여준다.
-w : 매 블로크에 몇가지 패턴으로 쓰기, 읽기를 하면서 검사한다.

매 디스크의 불량블로크를 검사하는 지령이다.

아래의 실례에서는 체계의 /dev/hda4디스크에 대한 불량블로크를 검사하였다. 이때 블로크수를 알자면 먼저 fdisk를 실행하여 디스크정보를 확인해야 한다. 블로크개수는 다음과 같은 방법으로 확인할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# fdisk -l /dev/hda

Disk /dev/hda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

 Device    Boot    Start       End    Blocks   Id  System
/dev/hda1   *          1       382   3061768+   b  Win95 FAT32
Partition 1 does not end on cylinder boundary.
/dev/hda2           1021      1532   4105080    c  Win95 FAT32 (LBA)
/dev/hda3           1532      4865  26777520    f  Win95 Ext'd (LBA)
/dev/hda4            382      1021   5133240   83  Linux
Partition 4 does not end on cylinder boundary.
/dev/hda5           1532      4810  26331448+   b  Win95 FAT32
/dev/hda6           4810      4865    446008+  82  Linux swap
```

우에서 실행한 지령의 결과로 /dev/hda4에는 5133240개의 블로크가 있다는것을 알수 있다.

주의 파일체계가 있는 장치에 -w추가선택을 사용하면 파일체계가 삭제되고 자료도 함께 손실되므로 자료가 있는 장치에 사용해서는 안된다.

```
root@Dolphin:~
[root@Dolphin root]# badblocks -v /dev/hda4 5133240
Checking for bad blocks in read-only mode
From block 0 to 5133240
Pass completed, 0 bad blocks found.
```

우에서 보는 바와 같이 /dev/hda4구획에는 불량블로크가 없다는것을 확인할수 있다.

불량블로크 (Bad Block)

디스크의 물리적손상을 말한다. 디스크에는 블로크를 단위로 하여 정보가 저장되게 되는데 충격 등으로 불량블로크가 생길수 있다. 지금 나오는 하드디스크들은 대부분이 자체로 불량블로크를 검사하여 관리한다.

Linux에서는 파일체계를 생성할 때 불량블로크에 관한 정보를 초기화하여 가지게 되는데 불량블로크가 많이 생기게 되면 분구를 나누어 일정한 부분을 사용하지 않을수도 있다.

블로크 (Block)

블로크란 파일체계가 련속적으로 할당하려고 하는 자료의 크기를 말한다. 실례로 파일체계의 블로크크기가 8KB라면 8KB까지의 파일은 디스크의 여러 부분에 나뉘여 있지 않고 항상 물리적으로 같은 자리에 련속적으로 존재하게 된다.

블로크의 크기가 크면 파일체계에 접근하는 성능은 좋아지지만 블로크 한개의 크기보다 작은 크기의 파일도 한개 블로크를 차지하므로 용량의 효률은 떨어진다. 또한 블로크의 크기가 작다면 공간랑비는 최소화할수 있으나 성능이 떨어지는 점이 있다. Unix나 Linux에서는 보통 512byte크기의 블로크를 사용한다.

basename

/bin/basename

파일이름추출

지령형식: basename 경로+파일이름[확장자]

경로+파일이름 : 경로를 포함한 파일이름.
확장자 : 추가선택으로 확장자까지 제거하고싶을 때에만 사용한다. 이 확장자가 파일이름의 맨 마지막에 오는 문자렬일 경우 삭제한다.
--help : 도움말을 보여준다.
--vesion: 판본정보를 보여준다.

이 지령은 파일의 완전경로를 보여주는 pwd지령과는 달리 경로를 포함한 파일이름을 인수로 받아 파일의 경로를 제거하고 필요에 따라서는 확장자까지 삭제하여 파일이름만을 보여준다.

/etc/issue.net의 파일경로와 확장자를 지워보자.

스크립트를 작성할 때 어떤 변수가 경로를 포함한 파일이름을 나타낸다고 생각해보자. 이 변수에서 파일이름만 추출하여 사용하고 싶을 때 basename을 리용하면 편리하다.

체계변수인 MAIL은 등록가입한 사용자의 우편파일과 경로를 저장한다. 등록가입한 사용자이름을 출력할 때에는 다음과 같이 할수 있다.

이런 점을 응용하면 스크립트를 만들 때 파일이름만을 추출하여 쉽게 사용할수 있다.

련관지령

dirname 경로-파일이름에서 경로이름만 추출한다.

batch

/usr/bin/batch

지령실행예약

지령형식: batch [-V] [-q 대기렬이름] [-f 파일이름] [-mv] [시간]

-q queue : 대기렬이름을 지정한다. 대기렬이름으로는 a~z, A~Z가 쓰일수 있다. 대기렬이름을 지정하지 않으면 at는 a를, batch는 b를 사용한다. 자모순서로 뒤의 이름을 가지는 대기렬은 더 큰 대기시간을 가진다. "="대기렬은 현재 수행되는 작업에 대한 대기렬로 예약되여있다. 대문자이름의 대기렬에 추가된 작업은 batch의 작업처럼 처리된다.

-m : 출력결과가 없더라도 작업이 완료될 때 사용자에게 우편을 보낸다.

-f filename : 스크립트파일 등을 실행할 때 사용한다.

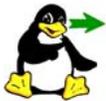
-v : 작업이 수행될 시간을 보여준다.

batch명령은 체계의 평균부하가 0.8이하로 내려가면 예약해둔 내용을 실행한다. 체계에 부하가 많이 걸리는 지령을 실행할 때 사용할수 있다.

먼저 현재체계의 평균부하를 살펴보자.

주의 체계의 평균부하는 1min, 4min, 15min의 평균을 보여준다.

현재평균부하가 2.51정도이라는것을 확인할수 있다. 체계의 부하가 적을 때 실행할 스크립트지령을 batch를 리용하여 등록한다. 등록후 atq지령을 사용하여 예약된 작업목록을 확인할수 있다.

등록된 시간을 보고 예약한 내용을 확인할수 있다.

련관지령

at	지정한 시간에 지령실행을 예약한다.
atq	예약된 작업목록을 보여준다.
atrm	예약된 작업목록을 삭제한다.
cron	정기적으로 예약된 작업을 수행한다.
batch	체계의 부하가 일정한 한계이하일 때 지령을 수행한다.

bc

/usr/bin/bc

본문형수산기

지령형식: bc [-lws] [-파일...]

-l : 표준수학서고를 정의한다.
w : POSIX bc에 대한 확장기능에 대하여 경고통보문을 보여준다.
-s : POSIX bc언어로만 실행하게 한다.

말단기에서 사용하는 대화형계산기언어이다.

bc는 대화형으로 문장을 해석하고 무한정확도의 수자를 지원해주는 언어이며 일반적으로는 수산기로 사용한다. C언어에서 사용하는 연산과 비슷하게 사용할수 있으며 지령행 추가선택을 주면 표준수학서고의 함수도 사용할수도 있다.

그러나 수산기로 사용할 때에는 계산하는 순서대로 식을 입력하여 답을 얻어낼수 있다.

실례로 900,000원의 돈을 저금하였을 때 년간 리윤률을 7%로 하여 5년이면 얼마가 되는가를 계산해보자. 리자를 계산하는 방법은 여러가지가 있겠지만 단순하게 원금*((1+리자)^기간)의 식으로 계산해보겠다.

식을 그대로 입력하기만 하면 계산결과를 얻을수 있다. 이외에도 bc는 일반수산기보다 정밀도가 훨씬 높다. 작업도중에 사용할수 있는것은 물론 복잡하면서도 정교한 계산도 할수 있다.

류동소수점과 고정소수점

류동소수점은 일반적인 실수를 가수(mantissa)와 지수(exponent)로 분리시켜 나타낸것이다. 이 방식은 한정된 비트수로 정밀도가 높은 계산을 하기 위한것으로서 가수는 수의 정밀도를, 지수는 수의 실제크기를 표현한다.

류동소수는 IEEE 754규격을 통해 옹근수부분과 지수부분으로 나뉘여 보관된다. 이때 전체 62bit중 첫 비트에는 부호를, 2번째에서 11번째까지 비트에는 10의 제곱수를, 나머지는 유효수자를 저장한다.

반면에 고정소수점은 일반적으로 사용하는 실수처럼 수를 옹근수부와 소수부로 나누어 사용한다. 화상처리나 도형처리연산에서는 대단히 많은 연산을 하게 된다. 그러나 앞에서 본 류동소수점연산은 매우 복잡하여 실수형수자를 저장하는 속도가 뜨므로 빠른 방법으로 실수를 처리하기 위해 등장한 방법이 바로 고정소수점연산이다.

이 연산은 옹근수부와 소수부로 나누어 각각 옹근수형태로 연산할수 있기때문에 빠른 속도로 처리할수 있다.

Apache봉사기의 정보를 숨기기

해당한 IP에서 실행하는 Apache봉사기의 정보는 보통 telnet IP 80으로 접속하여 알아볼수 있다. 이 웨브봉사기의 정보를 막아보자. Apache봉사기의 설정파일인 httpd.conf파일에서 아래의 ServerTokens지시자를 살펴보고 이것을 prod로 변경하면 된다.

```
# ServerTokens OS
ServerTokens Prod
```

변경한 다음에는 httpd데몬을 다시 시작하여 검사해본다.

biff

/usr/bin/biff

우편도착알림

 지령형식: biff [추가선택]

y : 우편도착을 통지해준다.
n : 우편도착을 통지하지 않는다.

우편봉사기를 일정한 간격으로 감시하여 새 우편이 도착하면 신호한다.

이 명령은 sendmail의 MTA를 사용할 때 유용하게 쓸수 있다. biff의 설정이 y로 되여있으면 새 우편이 도착했을 때 우편을 보낸 사람과 우편의 첫 부분이 화면에 나타난다. 그러면 현재 biff의 상태를 보도록 하자.

대부분의 체계에서는 biff가 n으로 설정되여있다. 이것을 y상태로 만들면 앞으로 우편이 올 때마다 통지받을수 있다. y상태로 만들고 확인해보자.

y 대신 yes를 사용할수도 있다.

주의 지금은 biff를 거의 사용하지 않는다. 대신 xbiff 등의 GUI응용프로그람이나 우편알림애플레트 등을 사용한다.

bzip2

/usr/bin/bzip2

파일압축

 지령형식: bzip2 [추가선택] [파일이름 ...]

-c, --stdout : 파일의 압축정보를 표준출력으로 내보낸다.
-d, --decompress : 압축을 푼다.
-z, --compress : 파일을 압축한다.
-t, --test : 압축된 파일에 대한 완전무결성을 검사하며 압축은 풀지 않는다.
-f, --force : 압축하거나 해제할 때 같은 이름의 파일이 있을 경우 덧쓰기한다.
-k, --keep : 파일을 압축하거나 해제할 때 원본파일을 지우지 않는다.
-s, --small : bzip2가 사용하는 기억기의 용량을 줄인다. bzip2작업에 2.5B만을 사용한다. 이렇게 작업하게 되면 속도는 절반이하로 떠지지만 2.5B의 작은 용량을 가지고도 bzip2를 사용할수 있다.
-q, --quiet : 오유가 나타나도 경고통보문을 내보내지 않는다.
-v, --verbose : bzip2작업진행상태를 자세히 볼수 있다
-L, --license, -V, --version : 사용허가정보와 판본정보를 보여준다.
-1 ~ -9 : 압축할 때의 블로크의 크기를 정해준다.

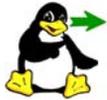
bzip2는 Burrows-Wheeler블로크의 정렬을 검사하는 압축알고리듬과 Huffman 부호화를 사용하는 압축률이 높은 지령이다. bzip2는 GNU gzip와 사용법이 비슷하다. 압축된 파일은 압축하기 전의 파일속성을 그대로 넘겨받는다. 그러므로 압축된 파일을 풀었을 때에도 원본파일의 속성이 그대로 보존된다. bzip2의 확장자는 .bz2이다.

우의 두 파일을 bzip2지령으로 압축해보자.

파일압축

파일압축 해제

bzip2 -d추가선택으로 파일압축을 해제한다.

bzip -d추가선택은 bunzip2지령과 같은 지령이다.

련관지령

compress　압축편의프로그람. (확장자 .Z)

gzip　표준 GNU/UNIX압축편의프로그람. (확장자 .gz)

Huffman부호화

자료코드를 통계적으로 검사하여 압축하는 방식이다. 자주 반복되는 코드는 적은 수의 bit로 표현하고 자주 반복되지 않는 코드는 상대적으로 많은 bit로 표현하여 압축한다.

bzip2압축파일이 손상되였을 경우

bzip2recover를 사용하여 손상된 압축파일을 복구한다.

```
bzip2recover [손상된 압축파일이름]
```

복구된 파일이 rec0001test.txt.bz2로 만들어진것을 확인할수 있다. rec0001test.txt.bz2의 완전무결성을 -t, -v추가선택으로 검사한다.

손상된 압축파일이 정상적으로 복구되였다는것을 확인할수 있다.

cal

/usr/bin/cal

달력보기

 지령형식: cal [추가선택] [[month] year]

-j : 달력을 보여준다.
-y : 현재년도의 달을 모두 보여준다.
-V : 판본정보를 보여준다.

체계에서 달력의 기능을 수행하는 지령이다.

사용할수 있는 년도는 원년부터 9999년까지이다. 1999년을 보려면 cal 99가 아니라 cal 1999라고 입력해야 한다. 아무런 지령행파라메터가 없으면 체계의 현재달을 보여준다.

년도와 달을 지정하여 해당한 달력을 볼수도 있다. 1979년 2월달을 보도록 하자.

월을 표시하지 않고 2005년의 전체 달력을 보고싶을 때에는 다음과 같이 한다.

```
[root@Dolphin Temp]# cal 2005
                              2005

       January               February                 March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                   1          1  2  3  4  5          1  2  3  4  5
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    6  7  8  9 10 11 12
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   13 14 15 16 17 18 19
16 17 18 19 20 21 22   20 21 22 23 24 25 26   20 21 22 23 24 25 26
23 24 25 26 27 28 29   27 28                  27 28 29 30 31
30 31
        April                   May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2    1  2  3  4  5  6  7             1  2  3  4
 3  4  5  6  7  8  9    8  9 10 11 12 13 14    5  6  7  8  9 10 11
10 11 12 13 14 15 16   15 16 17 18 19 20 21   12 13 14 15 16 17 18
17 18 19 20 21 22 23   22 23 24 25 26 27 28   19 20 21 22 23 24 25
24 25 26 27 28 29 30   29 30 31               26 27 28 29 30

        July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2       1  2  3  4  5  6                1  2  3
 3  4  5  6  7  8  9    7  8  9 10 11 12 13    4  5  6  7  8  9 10
10 11 12 13 14 15 16   14 15 16 17 18 19 20   11 12 13 14 15 16 17
17 18 19 20 21 22 23   21 22 23 24 25 26 27   18 19 20 21 22 23 24
24 25 26 27 28 29 30   28 29 30 31            25 26 27 28 29 30
31
       October               November               December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                   1          1  2  3  4  5                1  2  3
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   11 12 13 14 15 16 17
16 17 18 19 20 21 22   20 21 22 23 24 25 26   18 19 20 21 22 23 24
23 24 25 26 27 28 29   27 28 29 30            25 26 27 28 29 30 31
30 31
```

이 내용을 한번에 인쇄기로 인쇄하여 1년 달력으로 만들고싶으면 다음과 같이 입력하면 된다.

```
# cal 2005 | lpr
```

련관지령

date 　　오늘의 날자 혹은 지정한 날의 날자를 주어진 형식대로 출력한다.

cardctl

/sbin/cardctl

PCMCIA기판관리

지령형식: cardctl [추가선택] 지령

status [소케트번호] : 현재의 소케트상태기발을 보여준다.
config [소케트번호] : 현재의 소케트설정을 보여준다.
ident [소케트번호] : 망기판설정정보를 보여준다.
suspend [소케트번호] : 체계를 완료하고 전원을 끈다.
resume [소케트번호] : 소케트에 전원을 넣고 재설정한다.
reset [소케트번호] : 소케트에 재설정신호를 보낸다.
eject [소케트번호] : 준비된 다음에 망기판을 제거한다.
insert [소케트번호] : 준비된 다음에 망기판을 삽입한다.
scheme [소케트번호] : 분류표이름이 지정되지 않으면 현재 PCMCIA설정분류표를 보여준다.
분류표이름이 지정된다면 모든 PCMCIA장치를 새로운 분류표에 맞게 재설정한다.
-V : 판본정보를 보여준다.
-c config : /etc/pcmcia대신 지정한 등록부에서 설정자료기지와 설정스크립트를 불러온다.
-f scheme : /var/lib/pcmcia/scheme대신 지정한 파일에 현재설정분류표를 저장한다.

노트형콤퓨터는 물론 일반체계에서도 PCMCIA기판설정을 관리하는 지령이다.
다음의 화면은 cardctl지령에 사용하는 설정등록부이다.

```
[root@Dolphin root]# ls -F /etc/pcmcia/
cis/          ide*          network*      serial*
config        ide.opts      parport*      serial.opts
config.opts   isdn*         parport.opts  shared
ftl*          memory*       scsi*         wireless*
ftl.opts      memory.opts   scsi.opts     wireless.opts
```

config파일에서 설정할수 있는 PCMCIA기판목록들을 볼수 있다.

```
[root@Dolphin root]# cat /etc/pcmcia/config | more
#
# PCMCIA Card Configuration Database
#
# config 1.154 2001/01/05 00:03:17 (David Hinds)
#

# config.opts is now included at the very end

#
# Device driver definitions
#
device "3c589_cs"
  class "network" module "3c589_cs"

device "avm_cs"
  class "isdn" module "avm_cs"

device "avma1_cs"
  class "isdn" module "net/slhc", "misc/isdn", "misc/hisax"
opts "type=26 protocol=2", "avma1_cs"
--More--
```

cardctl status지령을 사용하면 PCMCIA기판의 상태를 확인할수 있다. 만일 인식한 PCMCIA기판이 없다면 다음과 같은 통보문이 나온다.

제대로 설치되였다면 현재의 기판상태와 정보를 보여준다. config지령으로는 현재 설정상태를 볼수 있으며 ident지령으로는 기판설정정보와 함께 제품정의, 제작회사의 고유코드 등을 볼수 있다.

PCMCIA

PCMCIA란 Personal Computer Memory Card International Association의 략자로서 개인용콤퓨터기억기카드국제협회를 말한다.

개인용콤퓨터에 사용되는 각종 망기판제품의 표준규격을 다루기 위해 1989년에 구성된 국제적인 규격제정협회이다.

PCMCIA기판은 간단한 자료저장장치로부터 정교한 기술을 요구하는 분야에까지 다양하게 쓰인다. 콤퓨터를 통신장치와 련결할 때도 리용되고 GPS체계 등에서도 리용되며 최근에는 휴대형장치에도 사용되고있다.

cardmgr

/sbin/cardmgr

PCMCIA장치관리자

지령형식: cardmgr [추가선택]

-V : 판본정보를 보여준다.
-q : 기판을 삽입할 때 체계음을 내지 않는다.
-v : 정상적인 동작을 활동안 상세한 정보를 보여준다.
-d : 구동프로그람모듈을 적재할 때 관련모듈의존성이 따라온다.(cardmgr는 insmod지령대신 modprobe지령을 사용한다.)
-f : 전경(foreground)방식으로 실행된다.
-o : 기판설정후 빠져나온다.(one pass)
-c configpath : /etc/pcmcia대신 지정한 등록부에서 설정자료기지와 설정스크립트를 불러온다.
-m modpath : /lib/modules/핵심부판본 등록부대신 지정한 등록부에서 핵심부모듈을 적재한다.
-p pidfile : /var/run/cardmgr.pid대신 지정한 파일에 cardmgr 프로쎄스 PID를 저장한다.
-s stabfile : /var/lib/pcmcia/stab대신 지정한 파일에 현재소케트정보를 저장한다.

PCMCIA기판에 대한 설정과 모듈관리를 담당하는 지령이다.

cardmgr지령은 /etc/rc.d/init.d/pcmcia데몬과 관련이 있다. /sbin/cardmgr는 소케트의 기판삽입과 제거를 조사하여 삽입이 된다면 /etc/pcmcia/config안의 기판정의목록을 살펴보고 적절한 모듈을 올려준다. 대체로 데몬만 실행해주면 스크립트는 실행상 오유인 PCMCIA기판을 자동인식한다.

PCMCIA기판인식여부는 cardctl ident지령으로 확인할수 있다.

만일 제대로 인식하지 못한다면 /etc/pcmcia/config파일을 열고 체계에 맞게 편집해야 한다. 설정이 끝나면 pcmcia데몬을 띄운다.

```
# /etc/rc.d/init.d/pcmcia restart
```

PCMCIA와 관련있는 파일은 다음과 같다.

/etc/pcmcia/config	기판설정자료기지
/etc/pcmcia/config.opts	PCMCIA장치를 위한 국부원천설정
/var/run/cardmgr.pid	현재 cardmgr프로쎄스 PID
/var/lig/pcmcia/stab	매 소케트에 대한 현재기판과 장치정보

cat

/bin/cat

파일보기

지령형식: cat [추가선택] [파일]...

-A, --show-all : -vET와 같다.
-b, --number-nonblank : 매 행의 맨 앞에 1부터 시작하는 행수를 표시한다.(빈 행은 제외)
-e : -vE와 같다.
-E, --show-ends : 매 행의 끝에 $를 붙여 보여준다.
-n, --number : 모든 행의 앞에 1부터 시작하는 행수를 표시한다.
-s, --squeeze-blank : 빈 행이 련속적으로 여러개 있는 경우 한행만 빈 행으로 보여준다.
-t : -vT와 같다.
-T, --show-tabs : TAB문자를 ^I로 보여준다.
-V, --show-nonprinting : 행시작문자와 TAB문자를 제외하고 ^I와 M표시를 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

cat는 본문파일의 내용을 보여주는 대표적인 지령이다. 이 지령은 파일내용을 단번에 보여주므로 내용이 많으면 보기가 힘들다. 이때에는 more지령이나 less, tail과 같은 지령을 사용하거나 출력내용을 흐름선을 통해 다른 프로그람과 조합하여 사용해야 한다.

MyWord.txt파일을 한번 보자. 행이 길어서 한행인지 두행인지 구별이 가지 않는다.

-n추가선택은 행앞에 번호를 붙여주는 역할을 한다. 아래의 화면에서 보는것처럼 이 파일은 총 6개 행이다.

주의 단순히 몇행으로 되였는가를 알기 위해서라면 wc -l지령을 사용한다.

행수가 많은 경우 흐름선으로 more지령을 실행하여 한페지씩 볼수 있다. 다음 페지를 보려면 공백건을 누르고 한행씩 내려가려면 Enter건을 누른다. 물론 more지령만으로도 같은 결과를 볼수 있다.

```
# cat /etc/httpd/conf/httpd.conf | more
```

주의 흐름선은 앞지령의 표준출력을 뒤지령의 표준입력으로 넘겨주는 역할을 한다.

또한 cat지령은 간단한 행편집기의 역할도 한다. 그러나 현재 입력하고있는 행은 수정할수 있지만 이미 지나온 행은 수정할수 없는 결함이 있다.

사용방법은 다음과 같다.

```
# cat > [파일이름]
```

그러면 실제로 파일을 만들어보자. 파일내용을 다 쓴 다음에는 [Ctrl+D]를 눌러 cat지령을 완료한다.

방금 입력한 편지가 정확히 보관되였는가를 보기 위하여 새로 만들어진 say.txt파일을 읽어보자.

파일의 내용을 화면에 출력하지 않고 직접 파일에 써넣을수도 있다. 다음의 실례는 song.txt파일의 내용을 say.txt파일에 써넣는다.

앞에서 작성한 say.txt의 내용이 지워지고 song.txt의 내용이 보관된다. 이미 보관된 본문의 뒤에 내용을 추가하려면 ">"를 두개 사용한다.

```
# cat song.txt >> say.txt
```

련관지령

more　　한페지씩 파일을 볼수 있다.
tail　　파일의 끝부분부터 지정한 행수만큼 보여준다.
less　　more와 같은 기능을 하지만 좀 더 추가적인 기능들이 있다.

cd

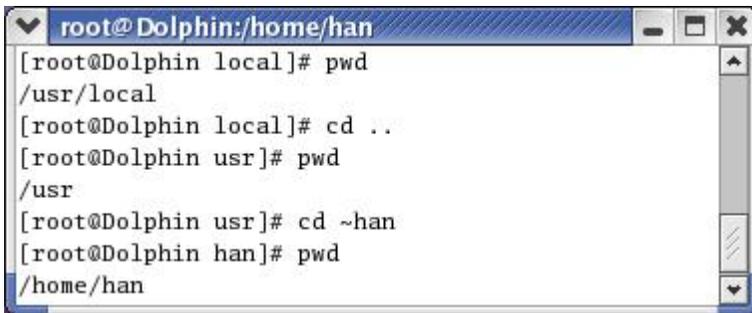
내부지령

등록부이동

지령형식: cd [등록부]

등록부를 이동하는 지령이다.

지 령	설 명
cd 등록부	이동하려고 하는 등록부로 이동한다.
cd .	현재등록부
cd ..	상위등록부로 이동한다.
cd $변수명	변수에 지정된 등록부로 이동한다.
cd /	가장 상위등록부로 이동한다.
cd ~|cd $HOME|cd	사용자의 home등록부로 이동한다.
cd ~사용자명	지정된 사용자의 home등록부로 이동한다.

pwd로 현재등록부를 확인한다. 한단계 우의 상위등록부로 이동하거나 사용자계정의 home등록부로 이동할수 있다.

cfdisk

/sin/cfdisk

구획설정

 지령형식: cfdisk [추가선택] [장치]

-a : 현재구획을 유표로 선택한다.
-c cylinders : 실린더개수를 지정한다.
-h heads : 머리부수를 지정한다.
-s sector-per-track : 자리길당 분구수를 지정한다.
-z : 구획정보를 읽지 않고 시작한다.
-P format : r(raw data), s(sector order), t(raw format)의 형식에 따라 구획표를 보여준다.

cfdisk는 지령행입력방식의 구획설정지령이다. Linux에는 기본구획설정지령인 fdisk외에 cfdisk나 설치시에 볼수 있는 Disk Druid 같은 도형방식의 프로그람이 있다.

주의 자료가 있던 장치에 새로 구획을 설정한 다음 재기동하면 원래의 구획표가 삭제되므로 거기에 있던 자료를 사용할수 없게 된다.

cfdisk지령을 실행한 다음에는 다음과 같은 지령들을 사용할수 있다.

?	도움말
b	선택된 구획으로 기동할수 있도록 설정
d	선택된 구획을 삭제
g	전문가방식
h	도움말
n	새로운 분구의 생성
p	현재의 분구정보를 출력
q	완료
w	보관

련관지령

fdisk 구획을 설정하기 위한 지령

MBR(Master Boot Record)

하드디스크구획정보를 담고있는 하드디스크의 첫번째 분구이다. MBR는 콤퓨터가 처음 기동될 때 BIOS가 읽어들이는 분구이다. 여기에는 어떤 분구가 기동가능한 분구인가 하는 분구정보와 기동분구를 읽어들이는 프로그람이 포함되여있다.

chage

/usr/bin/chage

통과어의 유효기일 지정

지령형식: chage [추가선택] 사용자이름

-m [limit] : 통과어가 유효한 최소날자를 설정한다. 0이면 매번 통과어를 변경해야 한다.
-M [limit] : 통과어가 유효한 최대날자를 설정한다.
-d [date] : 통과어의 마지막 변경날자를 YYYY-MM-DD형태로 설정한다.
-E [date] : 사용자계정에 더 이상 접근할수 없는 날자를 설정한다.
-I [date] : 계정이 잠금상태가 되기 전에 통과어가 만기가 되는 날자를 설정한다.
-W [date] : 통과어가 만기가 되기 전에 경고통보문을 내보내는 날자를 설정한다.
-l : 사용자통과어의 만기정보를 보여준다.

체계보안을 위해 사용자통과어의 유효기일을 설정 및 변경하는 지령이다.

이것은 /etc/passwd와 /etc/shadow파일을 참조한다.

체계보안에서 통과어관리는 매우 중요하다. 하지만 사용자가 너무 많으면 체계관리가 복잡해질수 있다. chage지령은 미리 지정한 날자에 의해 그 날자가 지나면 통과어를 변경하여 보다 효률적인 체계관리를 할수 있도록 도와준다.

먼저 han계정에 대한 통과어의 유효기일과 나머지정보를 보도록 하자.

```
[root@Dolphin root]# chage -l han
Minimum:        0
Maximum:        99999
Warning:        7
Inactive:       -1
Last Change:            Jan 26, 2005
Password Expires:       Never
Password Inactive:      Never
Account Expires:        Never
```

이제 han계정의 통과어가 유효하게 되는 최대날자수는 7일, 최소날자수는 1일, 경고날자는 7로 지정해보자.

-I추가선택에 0값을 설정하면 설정한 유효날자 이후에는 그 계정을 사용할수 없게 된다.

```
# chage -I 0 han
```

-E추가선택은 han계정을 2005년 4월 30일 이후 잠근다.

```
# chage -E 2005-04-30 han
```

han계정의 잠금을 해제시켜 본다.

```
# chage -E 0 han
```

chattr

/usr/bin/chattr

속성변경

지령형식: chattr [추가선택] 속성 파일...

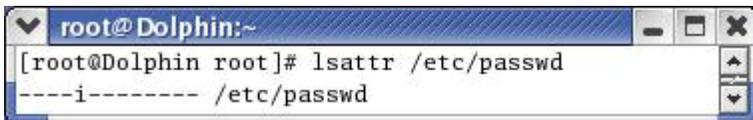
추가선택
-R : 등록부와 그 안의 부분등록부, 파일모두의 속성을 변경한다.
-V : 변경된 속성정보를 자세히 출력한다.
-v : 판본정보를 보여준다.
속성
a : 파일을 추가방식으로만 열수 있다
c : 핵심부에 의해 디스크상에 압축상태로 저장된다.
d : dump지령수행시 여벌복사되지 않는다.
D : 이 속성을 가진 등록부가 변경되면 변경되는 내용이 디스크에 동기적으로 반영된다.
I : 파일을 수정할수 없다. 오직 관리자만이 이 속성을 다시 변경할수 있다.
s : 파일이 지워질 때 일단 그 파일의 블로크들이 모두 0이 된 다음 디스크에 기록된다.
S : 파일이 수정될 때 그 변화가 디스크에 반영된다.
u : 파일이 지워지면 내용이 보관된다.

파일이 파괴되지 않도록 파일속성을 변경하는 지령이다. 실수로 보호하여야 할 파일들을 지우거나 덧쓰는 현상을 없애기 위해 리용한다.

"+"속성은 파일에 속성을 추가하고 "-"속성은 파일에서 속성을 제거한다. "="속성은 파일이 주어진 속성만을 가지도록 한다.

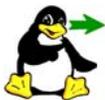
chattr +i추가선택으로 /etc/passwd에 속성을 부여해보자. 이렇게 하면 파일을 지울수도 이름을 변경할수도 내용을 추가할수도 련결을 생성할수도 없게 된다.

```
# chattr +i /etc/passwd
```

lsattr지령으로 변경된 속성들을 확인할수 있다.

"+i"속성이 부여된 /etc/passwd파일을 삭제해보자. 속성을 제거하기 전에는 이 파일을 삭제할수 없다. 마찬가지로 파일내용변경도 불가능하다. vi편집기로 파일을 열어도 읽기전용이므로 수정할수 없다.

주의 경험이 적은 관리자는 파괴자가 침범하여 설치하고 나간 뒤에 이 속성이 걸려있을 경우 삭제하지 못할수도 있다. 수상한 파일이 관리자의 권한으로도 삭제하거나 수정할수 없다면 반드시 lsattr를 사용하여 확인해야 한다.

이 "-i"속성을 /etc/passwd파일에서 제거해보자.

이제는 이 파일을 수정하거나 제거할수 있다.

련관지령

lsattr　　Linux파일체계의 파일속성을 보는 지령

chfn

/usr/bin/chfn

사용자정보변경

지령형식: chfn [추가선택] 사용자이름

사용자이름 : 정보를 변경하고 싶은 사용자이름. 공백이면 자기의 계정정보가 변경된다.
-f, --full-name : 사용자의 옹근이름을 변경한다. (사용자계정과 다름)
-h, --home--phone : 사용자의 집전화번호를 변경한다.
-o, --office : 사용자의 직장이름을 변경한다.
-p, --office-phone : 사용자의 직장전화번호를 변경한다.
-u, --help : 도움말을 보여준다.
-v --version : 판본정보를 보여준다.

passwd파일 등을 변경하지 않고도 finger지령을 사용했을 때 볼수 있는 사용자정보를

변경하는 지령이다.

관리자가 계정으로 등록된 사용자의 정보를 변경할 때 사용할수 있다. chfn지령은 /etc/passwd에 저장된 정보를 변경하므로 본문편집기에서 이 파일을 변경해주어도 된다.

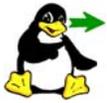
주의 관리자가 아닌 일반사용자는 /etc/passwd를 직접 수정할수 없다.

사용자정보는 finger지령으로 확인할수 있다.

관리자가 사용자의 정보를 변경하려면 다음과 같이 한다. 여기에서는 han이라는 사용자의 정보를 바꾸어보자.

사용자정보중에서 이름만 "HanUnJong"으로 변경할 경우에는 다음과 같이 할수 있다.

finger지령으로 변경된 정보를 확인한다.

직장이름이나 집전화번호 등도 마찬가지로 변경할수 있다.

련관지령

finger　사용자정보확인
chgrp　파일의 그룹변경
chmod　파일의 권한변경
chown　파일의 소유자변경
chsh　쉘변경

chgrp　　/bin/chgrp

그룹변경

지령형식: chgrp [추가선택] 그룹이름 파일이름

그룹이름 : 새로운 그룹이름
파일이름 : 그룹을 변경하고 싶은 파일이름
-c, --changes : 그룹이 변경되는 파일만 보여준다.
-f, --silent, --quiet : 그룹이 변경되지 않는 경우에도 오유통보문을 현시하지 않는다.
--help : 도움말을 보여준다.
-R, --recursive : 하위등록부에 있는 모든 등록부와 파일들의 그룹을 변경한다.
--reference=filename : 지정한 파일의 그룹을 변경한다.
-v, --verbose : 그룹변경을 자세히 보여준다.
--version : 판본정보를 보여준다.

파일의 소유그룹을 변경하는 지령이다. 주로 chmod지령과 함께 파일을 사용하는 접근권한을 설정할 때 사용한다. 새로운 그룹의 고유번호나 그룹이름으로 변경할수 있으며 /etc/group에서 새로운 그룹에 대한 정보를 확인할수 있다.

그룹의 변경은 파일의 소유자나 관리자만이 할수 있다. book등록부와 그안의 하위등록부를 모두 man그룹으로 바꾸고 변경된 그룹을 확인해보자.

기본그룹

/etc/group파일을 살펴보면 우리가 지정하지 않은 그룹들이 많다. 아래에 Linux를 설치하는 과정에 자동으로 설정되는 기본그룹들을 보여준다.

그 룹	GID	그룹성원
root	0	root
bin	1	root, bin, daemon
daemon	2	root, bin, daemon
sys	3	root, bin, adm
adm	4	root, adm, daemon
tty	5	
disk	6	root
lp	7	daemon, lp
mem	8	
kmem	9	
wheel	10	root
mail	12	mail
news	13	news
uucp	14	uucp
man	15	
games	20	
gopher	30	
dip	40	ftp
ftp	50	
nobody	99	
users	100	

이와 같은 기본그룹은 체계의 기본사용자와 관련되여있으므로 일정한 프로그람을 구동하기 위하여 사용되는 경우가 많다. 이것은 어떤 프로그람을 설치하는가에 따라 달라지지만 대부분의 경우 root와 deamon, sys, adm, disk, bin, wheel 등 root와 관련있는 그룹이 설치된다. 이와 같은 기본그룹들은 일반사용자들이 변경하지 못하도록 해야 한다.

chkconfig

/sbin/chconfig

구동프로그람설정

지령형식: chkconfig --list [이름]
chkconfig --del 이름
chkconfig [--level 준위] 이름 <on | off | reset>

체계를 기동할 때 자동실행할 봉사를 실행준위별로 보고 갱신하는 지령이다.

기동시 자동실행되는 봉사데몬은 /etc/rc.d등록부에 포함되여있다. rc뒤에 붙은 수자는 실행준위이다. 즉 rc0.d는 실행준위가 0, rc1.d는 1이라는 의미이다.

rc5.d등록부에서 K로 시작하는 파일은 체계를 기동할 때 봉사하지 않는 데몬이며 S로 시작하는 파일은 기동시 봉사할 데몬이다.

S다음의 수자는 봉사할 데몬의 순서이다.

```
root@Dolphin:~
[root@Dolphin root]# ls /etc/rc.d/rc5.d/
K05innd          K50snmpd        K84ripd         S55sshd
K05saslauthd     K50snmptrapd    K84ripngd       S56rawdevices
K12mailman       K50tux          K85zebra        S56xinetd
K15httpd         K50vsftpd       K95firstboot    S80sendmail
K15postgresql    K54pxe          S05kudzu        S80spamassassin
K20nfs           K65kadmin       S08iptables     S85gpm
K24irda          K65kprop        S09isdn         S90crond
K25squid         K65krb524       S10network      S90cups
K28amd           K65krb5kdc      S12syslog       S90xfs
K34dhcrelay      K70aep1000      S13portmap      S91smb
K34yppasswdd     K70bcm5820      S14nfslock      S95anacron
K35dhcpd         K74ntpd         S17keytable     S95atd
K35vncserver     K74ypserv       S20random       S97rhnsd
K35winbind       K74ypxfrd       S24pcmcia       S99local
K36lisa          K84bgpd         S25netfs
K45named         K84ospf6d       S26apmd
K46radvd         K84ospfd        S28autofs
```

기동할 때 K로 시작하는 데몬파일이름을 S로 변경할수도 있지만 chkconfig지령을 사용하면 손쉽게 할수 있으며 보다 구체적인 정보를 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# chkconfig --list | more
spamassassin    0:off   1:off   2:on    3:on    4:on    5:on    6:off
kudzu           0:off   1:off   2:off   3:on    4:on    5:on    6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
random          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rawdevices      0:off   1:off   2:off   3:on    4:on    5:on    6:off
pcmcia          0:off   1:off   2:on    3:on    4:on    5:on    6:off
saslauthd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
keytable        0:off   1:on    2:on    3:on    4:on    5:off   6:off
apmd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
atd             0:off   1:off   2:off   3:on    4:on    5:on    6:off
gpm             0:off   1:off   2:on    3:on    4:on    5:on    6:off
--More--
```

우의 목록에서 keytable데몬은 실행준위 5에서 해제되여있다. 이것을 능동으로 하자.

```
# chkconfig --level 5 keytable on
```

--list추가선택은 봉사정보를 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# chkconfig --list keytable
keytable  0:off  1:on  2:on  3:on  4:on  5:on  6:off
```

설정한 실행준위 5에서의 데몬봉사를 다시 해제해보자

```
root@Dolphin:~
[root@Dolphin root]# chkconfig --level 5 keytable off
[root@Dolphin root]# chkconfig --list keytable
keytable  0:off  1:on  2:on  3:on  4:on  5:off  6:off
```

--del지령으로 지정한 데몬봉사를 모든 실행준위에서 해제할수도 있다.

```
root@Dolphin:~
[root@Dolphin root]# chkconfig --del keytable
[root@Dolphin root]# chkconfig --list keytable
service keytable supports chkconfig, but is not referenced in
any runlevel (run 'chkconfig --add keytable')
```

그러면 다시 --add지령으로 0, 6준위를 제외한 모든 실행준위를 능동으로 하자.

chmod

/bin/chmond

호출권한변경

 지령형식: chmod [추가선택] 호출권한 파일이름

호출권한 : 새로운 호출권한
파일이름 : 호출권한을 변경하려는 파일이름
-c, --changes : 변경된 파일들에 대한 정보를 보여준다.
-f, --silent, --quiet : chmod로 변경되지 않는 파일에 대한 오유를 보여주지 않는다.
--help : 도움말을 보여준다.
-R, --recursive : 하위등록부와 파일모두를 다 변경한다.
--reference=filename : 지정한 파일을 참조하여 호출권한을 변경한다.
-v, --verbose : 매개 파일에 대하여 변경된 정보나 변경되지 않은 정보를 구체적으로 보여준다.
--version : 판본정보를 보여준다.

파일의 호출권한을 변경하는 지령이다. chmod지령은 그 파일의 소유자와 체계관리자만이 사용할수 있으며 파일의 소유자, 파일의 그룹, 다른 사용자로 나누어 권한을 설정할수 있다.

chmod를 리용한 파일 및 등록부의 호출권한변경에는 8진수를 리용한 변경법과 기호에 의한 변경법이 있으며 결과는 같다.

호출권한보기

파일이나 등록부는 모두 개별적으로 권한이 설정되여있다. 파일이나 등록부의 권한은 ls지령을 리용하여 볼수 있다.

호출권한은 등록부나 파일을 사용할수 있는 상태를 의미한다.

호출권한	설 명	8진수 표현
d---------	파일/등록부의 구분(파일:-, 등록부 : d)	
-r--------	소유자가 읽기권한이 있다.	400
--w-------	소유자가 쓰기권한이 있다.	200
---x------	소유자가 실행권한이 있다.	100
----r-----	그룹이 읽기권한이 있다.	40
-----w----	그룹이 쓰기권한이 있다.	20
------x---	그룹이 실행권한이 있다.	10
-------r--	다른 사용자가 읽기권한이 있다	4
--------w-	다른 사용자가 쓰기권한이 있다	2
---------x	다른 사용자가 실행권한이 있다	1

기호에 의한 파일권한변경

기호는 파일이나 등록부를 리용할수 있는 사용자를 구분하기 위한것이다. 소유자나 그룹, 그외의 사용자가 서로 다른 권한을 가지도록 파일이나 등록부에 권한을 설정할수 있다. 실레로 test.txt라는 파일을 다른 사용자가 읽을수 있도록 권한을 주자면 다음과 같이 입력한다.

```
# chmod o+r test.txt
```

test.txt파일에 소유자와 그룹에게는 읽기/쓰기권한을 주고 다른 사용자의 읽기권한은 금지시키려면 다음과 같이 입력한다.

```
# chmod ug+rw,o-r test.txt
```

이 경우 ","을 리용하여 단번에 설정할수 있다.

사용자기호

u	user	파일이나 등록부의 소유자
g	group	파일이나 등록부의 그룹
o	other	다른 사용자
a	all	소유자, 그룹, 다른 모든 사용자 (아무런 표시도 안할 경우에 기정으로 설정됨)

호출권한기호

r	read	파일/등록부에 읽기권한을 준다.
w	write	파일/등록부에 쓰기권한을 준다.
x	execute	파일/등록부에 실행권한을 준다.
s	set user (group) ID	파일실행시 파일의 소유자 혹은 그룹권한으로 실행된다.
t	sticky bit	점착비트를 설정한다.
u	user	현재소유자의 호출권한설정과 같은 내용으로 변경된다.
g	group	현재그룹의 호출권한설정과 같은 내용으로 변경된다.
o	other	현재 다른 사용자의 호출권한설정과 같은 내용으로 변경된다.
l	locking	강제로 파일을 잠근다.

설정기호

+	호출권한허가	지정한 호출권한을 허가한다.
-	호출권한금지	지정된 호출권한을 금지시킨다.
=	호출권한지정	지정된 호출권한만 허가하고 나머지는 금지시킨다.

8진수를 리용한 파일권한변경

네자리 8진수를 리용하여 파일이나 등록부의 권한을 변경할수 있다. 리용하는 8진수는 아래에 있는 표에서 보여준다.

파일/등록부소유자

0400	파일/등록부의 소유자에게 읽기권한을 준다.
0200	파일/등록부의 소유자에게 쓰기권한을 준다.
0100	파일/등록부의 소유자에게 실행권한을 준다.

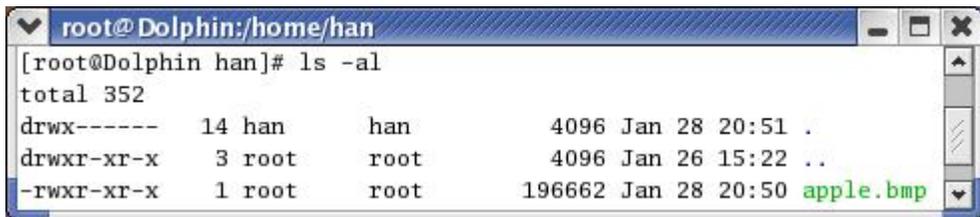
파일/등록부 그룹

0400	파일/등록부의 그룹에게 읽기권한을 준다.
0200	파일/등록부의 그룹에게 쓰기권한을 준다.
0100	파일/등록부의 그룹에게 실행권한을 준다.

다른 사용자

0004	다른 사용자에게 읽기권한을 준다.
0002	다른 사용자에게 쓰기권한을 준다.
0001	다른 사용자에게 실행권한을 준다.

4000	파일실행시 파일의 소유자 혹은 그룹권한으로 실행한다.
2000	파일실행시 파일의 소유자 혹은 그룹권한으로 실행한다.
1000	점착비트

파일정보를 보기 위해 ls를 사용한다.

주의 SetUID(4000), SetGID(2000)지령을 사용하면 실행파일의 내부에 쉘을 띄우는 코드가 있는 경우 보안상 문제가 생긴다. 꼭 필요한 경우가 아니라면 사용하지 않는것이 안전하다.

실례로 apple.bmp파일에 대한 권한을 소유자는 "읽기, 쓰기, 실행", 그룹은 "읽기, 쓰기", 다른 사용자는 "실행"으로 변경해보자.

먼저 기호방식으로 변경해보자.

현재 소유자는 읽기쓰기실행권한으로 되여있으므로 수정할 필요가 없으며 그룹은 읽기실행권한으로 되여있으므로 실행권한을 금지시키고 쓰기권한을 주어야 하며 다른 사용자는 읽기실행권한으로 되여있으므로 읽기권한을 금지시키면 된다. 즉

그룹(u)에는 실행(x)권한금지, 쓰기(w)권한주기: u-x+w

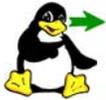
다른 사용자(o)에게는 읽기(r)권한금지: o-r

```
# chmod u-x+w,o-r apple.bmp
```

다음으로 8진수방식으로 변경해본다.

소유자에게는 읽기(400), 쓰기(200), 실행(100)권한을, 그룹에는 읽기(40), 쓰기(20) 권한을, 다른 사용자에게는 실행(1) 권한을 준다.

소유자 읽기, 쓰기, 실행: 400+200+100: 700
그룹 읽기, 쓰기: 40+20: 60
다른 사용자 실행: 1: 1

```
# chmod 761 apple.bmp
```

련관지령

chfn　사용자정보변경
chgrp　그룹변경
chown　소유자변경
chsh　쉘변경

SetUID와 SetGID

파일이나 등록부의 파일형식과 권한에 대한 내용은 해당 파일의 inode에 16bit로 보관되여있다. 이 중에서 1~12bit는 표에서 보는바와 같이 파일의 실행에 대한 권한이다.

파일 ID비트	파일실행비트	소유자권한비트	그룹권한비트	기타 권한비트
16 15 14 13	12 11 10	9 8 7 6	6 5 4	3 2 1

이중에서 10~12bit부분은 실행파일을 어떤 방식으로 실행시키겠는가를 지정한다. 이 부분은 앞서 말한 setuid(4000), setgid(2000), 점착(1000)비트로 표현한다. 이 비트는 파일의 속성이 등록부일 경우에는 무시된다.

실행파일이 실행되면 기억기에 적재되여 프로세스가 생성된다. 이 실행파일을 소유자가 아닌 다른 사람이 실행하다가 저장해야 할 경우가 있을수 있는데 이때 프로세스와 파일사이에 권한을 검사할수 있는 통로로 사용하는것이 EUID/EGID이다.

chown

/bin/chown

소유자변경

지령형식: chown [추가선택...] 소유자:[그룹]파일...
chown [추가선택...] 그룹파일...

-f, —silent, —quiet : 이 지령으로 변경되지 않는 파일들에 대해 오유통보문을 보여주지 않는다.
-R, --recursive : 하위등록부/파일에 모두 적용한다.
-v, --verbose : 변경되는 상태를 구체적으로 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일에 대한 소유자와 그룹을 지정한 소유자의 그룹으로 변경하는 지령이다.

Apache웨브봉사를 위한 index등록부를 한번 살펴보자. Red Hat패키지로 설치한 경우에 이것은 /var/www/html등록부로 된다.

현재 index.html의 소유자와 그룹은 모두 admin으로 되여있는것을 볼수 있다.

index.html의 소유자와 그룹을 모두 webmaster로 변경해보자.

```
# chown webmaster:webmaster/var/www/html/index.html
```

index.html파일의 소유자와 그룹이 모두 webmaster로 변경되였다.

소유자만 변경할 경우에는 다음과 같이 한다.

```
# chown admin index.html
```

index.html파일의 소유권을 확인해보자. 이 파일의 소유자는 admin으로, 그룹은 webmaster로 되였을것이다.

그룹만 변경할수도 있다. 이때는 점(.)으로 시작하는 그룹을 지정한다.

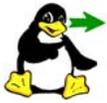
```
# chown .webmin index.html
```

주의 chgrp지령으로도 같은 작업을 수행할수 있다.

변경된 그룹을 확인해보자. 소유자는 admin그대로이며 그룹만 webmin으로 되였다.

/var/www/html을 포함하여 하위등록부/파일에까지 모두 소유권변경을 적용하려면 -R추가선택을 사용한다.

```
# chown -R webmaster:webmaster /var/www/html
```

결과는 다음과 같다.

련관지령

newgrp　현재 속해있는 그룹을 바꾼다.

chroot

/usr/sbin/chroot

가상뿌리등록부

지령형식: chroot 새로운 뿌리경로 [지령]
chroot [추가선택]

--help : 도움말을 보여준다.
--version : 지령에 대한 판본정보를 보여준다.

보안을 위해 새로운 가상뿌리등록부를 만드는 지령이다.

사용자가 실행하는 지령은 보통 뿌리등록부인 /을 기준으로 하는 $PATH에 따라 해석하게 된다. chroot지령은 뿌리등록부를 일반적인 환경과는 다르게 체계관리자가 지정한 등록부로 바꾸어준다. 이것은 체계관리나 보안에 아주 효과적이다.

실례로 웨브봉사기의 기본등록부인 /var/www/html을 체계의 뿌리등록부처럼 해놓으면 침해자를 속일수 있다. 이 설정에 관한 자세한 내용은 다른 참고서들을 참고하기 바란다.

주의 이 지령을 사용할 때에는 봉사를 사용하는데 필요한 서고와 실행파일을 모두 이 경로에 복사해야 한다.

chsh

/usr/bin/chsh

쉘변경

지령형식: chsh [추가선택] [계정이름]

-s shell : 지정하는 쉘을 앞으로 등록가입할 쉘로 바꾼다.
-l, --list-shells : /etc/shells파일안에 지정된 쉘을 라렬한다.
-u, --help : 도움말을 보여준다.
-v, --version : 판본정보를 보여준다.

등록가입하여 사용하는 기본 쉘에서 림시로 다른 쉘을 사용하게 하는 지령이다.

이것은 /etc/shells에 등록된 쉘만을 변경할수 있다. 만일 /etc/shells파일에 쉘이 존재하지 않는다면 오유통보문을 내보낸다.

먼저 현재체계에서 사용할수 있는 쉘을 살펴보자.

```
root@Dolphin:/home/han
[root@Dolphin han]# cat /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/bin/bash2
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh
```

chsh -l추가선택은 /etc/shells파일과 같은 내용을 보여준다.

```
root@Dolphin:/home/han
[root@Dolphin han]# chsh -l
/bin/sh
/bin/bash
/sbin/nologin
/bin/bash2
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh
```

그러면 chsh지령을 리용하여 현재쉘을 /bin/csh로 변경해보자.

-s추가선택을 리용하여 직접 지정한 쉘로 변경할수도 있다.

자기가 사용하는 쉘을 알아보기 위해서는 다음과 같이 한다.

```
# echo $SHELL
```

쉘의 종류

쉘은 체계호출, 열기, 읽기, 쓰기, 닫기 등의 기본기능으로 지령을 해석하여 프로쎄스를 생성하고 핵심부와 사용자사이를 련결하는 도구이다.

Linux에서 사용할수 있는 쉘에는 Bourne shell(sh), C shell(csh), TC shell(tcsh), Korn shell(ksh), Bourne Again Shell(bash)을 비롯하여 여러가지가 있다.

clear

Usr/bin/clear

화면지우기

지령형식: clear

말단기의 본문화면을 지우고 유표를 맨 웃줄로 올리는 지령이다. DOS의 cls지령과 같다.

아래와 같이 파일목록보기지령인 ls -a로 목록을 출력하고 화면에 가득찬 내용을 clear지령으로 지워보자.

cmp

/usr/bin/cmp

파일비교

지령형식: cmp [추가선택]

-l, --verbose : 파일들의 차이점을 구체적으로 보여준다.
-S, --quiet, --silent : 아무런 통보문도 보여주지 않는다.

diff와 비슷하지만 좀 더 간단한 지령이다. 두 파일을 비교하여 다른 부분을 알려준다. cmp지령은 diff지령의 간단한 판본이라고 생각할수 있다. diff가 두 파일사이의

차이점에 대해서 통보한다면 cmp는 단지 두 파일사이의 서로 다른 부분만을 보여준다.
query파일과 query2파일을 비교해보자. 두 파일은 첫번째 줄에서 하나의 글자가 차이난다.

col

usr/bin/col

행시작문자변환

지령형식: col [추가선택]

-b : 공백문자를 출력하지 않고 공백문자와 련결되는 마지막 문자만 출력한다.
-h : 여러 공백문자를 태브문자로 바꾼다.
-x : 일반적으로는 공백를 태브로 바꾸지만 이 추가선택은 여러 공백문자들을 그대로 둔다.
-l 수자 : 기억기에 한번에 둘수 있는 최대행수를 사용자가 지정한 수값으로 한다.(기정은 128행)

본문파일의 행바꾸기문자와 공백문자 등을 바꾸어주는 지령이다. 다시말하여 \n\r문자를 \n문자로, 공백문자를 태브문자로 바꾸거나 공백문자를 없앤다.
다음은 지도서페지를 받아서 파일로 보관하는 실례이다.

```
# man httpd | col > httpd.man
```

colcrt

/usr/bin/colcrt

밑줄문자변환

지령형식: clocrt [추가선택] [파일]

- : 밑줄(_)을 보여주지 않는다.
-2 : 밑줄이 있는 행에는 그 다음 행에 "-"표시를 하고 없는 행에는 빈 행을 추가한다.

밑줄문자를 감추거나 그 다음 행에 "-"표시를 하는 변환려과기능을 수행하는 지령이다.

실례를 통하여 보도록 하자.

현재 query라는 파일에는 밑줄문자가 4개 포함되여있다. date_format와 cst_users, cst_productregs 그리고 cst_productregdetail이다.

```
root@Dolphin:/home/han
[root@Dolphin han]# cat query
slect date_format(uregdate,'%Y%m') date,count(uid) from cst_
users u,cst_productregs p, cst_productregdetail d
where u.uno=p.preguno
and p.preguno=d.preguno
group by date
```

colcrt로 려과하여 밑줄문자가 한행아래에 "-"로 표시된것을 볼수 있다.

```
root@Dolphin:/home/han
[root@Dolphin han]# cat query | colcrt
slect date format(uregdate,'%Y%m') date,count(uid) from cst
users u,cst productregs p, cst productregdetail d
          -                                               -
           -                  -

where u.uno=p.preguno
and p.preguno=d.preguno
group by date
```

colcrt -추가선택은 밑줄문자를 보여주지 않는다.

root@Dolphin:/home/han

```
[root@Dolphin han]# cat query | colcrt -
slect date format(uregdate,'%Y%m') date,count(uid) from cst
users u,cst productregs p, cst productregdetail d
where u.uno=p.preguno
and p.preguno=d.preguno
group by date
```

colcrt -2추가선택을 리용한 실례는 다음과 같다.

root@Dolphin:/home/han

```
[root@Dolphin han]# cat query | colcrt -2

slect date format(uregdate,'%Y%m') date,count(uid) from cst
users u,cst productregs p, cst productregdetail d
            -                                             -
             -                   -
where u.uno=p.preguno

and p.preguno=d.preguno

group by date
```

colrm /usr/bin/colrm

지정한 범위의 문자렬 삭제

 지령형식: colrm [시작위치] [마감위치]

파일에서 선택된 범위의 문자렬을 삭제한다.

colrm지령의 파라메터가 한개의 수이면 지정된 위치부터 마지막까지의 문자렬을 삭제한다. 만일 시작위치와 마감위치를 지정하면 그 범위안에 있는 문자렬들을 모두 삭제한다.

실례를 들어보자. uname지령은 체계정보를 보여주는 지령이다. 처음에는 조작체계의 종류가, 다음에는 콤퓨터이름이 출력된다.

이제 colrm지령으로 콤퓨터이름만을 추출해보자.

앞에 있는 조작체계의 종류를 "colrm 1 6"지령으로 삭제한다. 여기서 1은 시작위치이며 6은 마감위치이다.

콤퓨터이름뒤에 있는 정보까지 삭제하려면 colrm 8지령을 추가한다. 결과 콤퓨터이름이 Dolphin이라는것을 알수 있다.

등록가입시에 보여주는 통보문 변경

일반적으로 주콤퓨터에 접속할 때 체계의 핵심부정보와 배포판정보를 보여준다. 이것은 보안상 약점으로 될수 있다.

등록가입할 때 나타나는 체계통보문은 /etc/issue파일에 보관되여있다.

등록가입할 때 보여주는 체계통보문만이 아니라 원격접속하는 콤퓨터에 대한 정보까지 포함하고있는 파일은 /etc/issue.net이다. 이 파일들을 사용자의 요구대로 변경할수 있다. 그러나 체계기동시 /etc/rc.d/rc.local파일에서 체계정보를 다시 불러들이기때문에 아래와 같이 /etc/issue.net에 대한 정보를 주석으로 처리해야 한다.

/etc/motd파일은 체계에 등록가입한 후에 표시되는 통보문정보를 보관한다.

```
echo "" > /etc/issue
echo "$R" >>/etc/issue
echo "Kernel $(uname -r) on $a $SMP$(uname -m)" >> /etc/issue
# cp -f /etc/issue /etc/isssue.net
```

column

/usr/bin/column

파일정렬

지령형식: column [추가선택] [파일...]

-c num : 전체 가로폭을 지정한 num값으로 한다.
-s char : -t추가선택을 위해 가로칸조사에 사용할 문자를 지정한다.
-t : 입력되는 내용의 가로칸개수를 조사하고 그것을 바탕으로 출력양식을 정한다. 입력되는 내용의 가로칸개수는 공백으로 구분한다.
-x : 가로로 먼저 라렬하고 그 다음에 세로로 라렬한다.

본문파일의 내용을 가로로 보기 좋게 정렬하여 보여주는 지령이다. 먼저 세로로 라렬하고 그 다음 가로로 라렬한다. 또한 태브구분자를 넣어 보기 좋게 할수도 있다.

```
root@Dolphin:/home/han
[root@Dolphin han]# (printf "PERM LINKS OWNER GROUP SIZE MONTH DAY HH:MM
NAME\n"; ls -l | sed 1d) | column -t
PERM        LINKS  OWNER  GROUP  SIZE    MONTH  DAY  HH:MM  NAME
-rw-r-x--x  1      root   root   196662  Jan    28   20:50  apple.bmp
drwxr-xr-x  2      root   man    4096    Jan    28   15:44  book
-rw-r--r--  1      root   root   3902    Jan    28   21:51  httpd.man
-rw-r--r--  1      root   root   39      Jan    28   21:46  query~
-rw-r--r--  1      root   root   211     Jan    27   15:09  test.txt
```

우의 실례에서는 ls -l지령으로 출력되는 현재등록부의 파일목록에 렬제목을 붙여 보기 쉽게 하였다. sed 1d지령으로 합계부분을 모두 삭제하고 printf지령으로 매개 마당에 대한 렬제목을 우와 같이 출력하였다. 이것을 column -t추가선택으로 받아 입력되는 내용의 가로칸개수를 조사하여 출력양식을 정하게 하였다.

comm

/usr/bin/comm

한행씩 비교

지령형식: comm [추가선택] 파일 1 파일 2

-1 : 파일 2를 기준으로 파일 1과 비교하여 같지 않은 부분을 보여주고 다음 렬에 같은 부분을 보여준다.
-2 : 파일 1을 기준으로 파일 2와 비교하여 같지 않은 부분을 보여주고 다음 렬에 같은 부분을 보여준다.
-3 : 파일 1과 파일 2를 비교하여 첫번째 렬에는 파일 1의 유일한 내용을, 두번째 렬에는 파일 2의 유일한 내용을 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

comm지령은 여러가지 목적에 리용되는 파일비교편의프로그람으로서 이 지령을 수행하기전에 비교할 파일들이 정렬되여있어야 정확하게 결과를 얻을수 있다.

실례를 통하여 보기로 하자. 먼저 sort지령으로 test1과 test2 파일을 정렬시키자.

다음의 화면은 -1추가선택을 사용한 결과이다. 파일 2를 기준으로 파일 1과 같지 않은 부분은 첫번째 렬에, 같은 내용은 두번째 렬에 출력된다.

-2추가선택을 사용해보자. 파일 1을 기준으로 첫번째 렬에는 파일 2와 같지 않은 부분을 보여주고 두번째 렬에는 같은 내용을 보여준다.

-12추가선택을 함께 리용하면 test1파일과 test2파일의 공통된 내용을 볼수 있는데 uniq지령을 함께 리용하면 렬에서 중복되지 않는 내용을 test3파일에 보관할수 있다.

```
# comm -12 test1 test2 | uniq > test3
```

cp

/bin/cp

파일복사

지령형식: cp [추가선택] 원본파일 복사파일
cp [추가선택] 파일... 등록부

-a, --archive : 원본파일의 속성, 련결정보들을 그대로 유지하면서 복사한다. -dpR추가선택과 같다.
-b, --backup : 복사대상파일이 이미 존재하는 경우 여벌파일을 만든다.
-d, --no-dereference : 원본파일이 기호련결파일이면 기호련결원본을 복사한다.
-f, --force : 복사대상파일이 있을 경우 복사대상파일을 강제로 지우고 복사한다.

-i, --interactive : 복사대상파일이 있을 경우 사용자에게 복사에 대한 여부를 물어본다
-l, --link : 복사하는것이 아니라 련결한다.
-P, --parents : 원본파일을 경로와 함께 지정했을 때 그 경로그대로 복사한다.
-p, --preseNer : 원본파일의 소유자, 그룹, 권한, 시간정보들이 그대로 보존되여 복사된다.
-r, -R, --recursive : 경로안에 있는 모든 하위등록부들과 파일들을 다 복사한다.
-s, --symbolic-link : 일반파일을 기호련결형식으로 복사한다.
-u, --update : 복사할 대상이 이미 있는 경우 그 파일의 변경날자가 원본파일과 같거나 더 최근의것이면 복사하지 않는다.
-v, --verbose : 복사상태를 구체적으로 보여준다.
-x, --one-file-system : 원본과 복사파일의 파일체계가 다를 경우에는 복사하지 않는다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일을 지정된 경로에 복사하는 지령이다.

복사할 경우 복사파일이 있으면 사용자에게 문의하지 않고 그대로 덧쓰기한다. 이때 원본파일의 복구는 불가능하다. 이와 같은 실수를 막기 위해 alias설정으로 -i추가선택을 설정하였다.

아래의 실례에서는 Apache설정파일과 이름봉사기설정파일을 home등록부에 복사한다.

```
# cp /etc/httpd.conf /etc/named.conf ~
```

-R추가선택은 웨브봉사기의 기본등록부인 /var/www/html과 모든 하위등록부, 그 안에 있는 파일들을 /backup등록부로 복사한다.

```
# cp -R /var/www/html /backup
```

inode

Linux나 UNIX에서 파일을 구별하는것은 파일이름이 아니라 inode라는 수자이다.

간단히 말해서 파일이름은 inode구조체표에 저장된 파일정보를 사람이 인식하기 쉽게 만든것에 지나지 않는다. 다음과 같은 실레를 통하여 구체적으로 확인해보자.

먼저 적당한 이름의 빈 파일을 하나 만들고 inode를 확인한다. mv지령을 사용하여 이 파일의 이름을 다른 이름으로 바꾸고 다시 inode를 확인해본다. 그러면 이 두 파일은 이름은 서로 다르지만 inode는 같다는것을 알수 있다. 이것은 우리가 어떤 이름을 사용하든지간에 체계는 두 파일을 같은 파일로 인식하고있다는것을 말해준다.

이번에는 cp지령을 리용하여 다른 등록부에 원래의 이름으로 파일을 복사하고 다시 비교해보자. 이때에는 inode가 달라졌음을 알수 있다. 비록 이름은 같지만 체계는 이 두 파일을 서로 다른것으로 인식하고있다.

cpio

/bin/cpio

복사판만들기

지령형식:

cpio -o, --create [추가선택] : 표준입력으로 받은 파일목록을 압축파일형태로 출력복사한다.

cpio -i, --extract [추가선택] [패턴] : 지정한 패턴의 형태에 맞는 압축파일로부터 입력을 받아 압축을 푼 파일로 복사한다.

cpio -p, --pass-through [추가선택] 등록부 : 같은 체계에서 지정한 다른 등록부로 파일을 복사한다.

-0, --null : 파일이름의 끝이 행시작문자가 아니라 빈 문자로 끝난다.
-a, --reset-access-time : 파일호출시간을 재설정한다.
-A, --append : 압축된 파일에 파일을 추가한다. -o추가선택에서만 동작한다.
-B : I/O블로크를 5120B로 지정한다. 기정값은 512B이다.
--block-size=BLOCK-SIZE : I/O블로크크기를 BLOCK-SIZE*512B로 지정한다.
-c : SVR4양식을 사용하기 위해 "-H newc"추가선택을 리용한다. 이전의 ASCII양식을 사용하려면 "-H odc"을 사용한다.
-C IO-SIZE, --io-size=IO-SIZE : I/O블로크크기를 지정한다.
-d, --make-directories : 지정한 등록부를 생성한다.
-E FILE, --pattern-file=FILE : -i추가선택에서 사용하며 지정한 형태의 형식에 맞는 파일을 읽는다.
-f, --nonmatching : 주어진 패턴에 상관없이 파일을 복사한다.

-F, --file=archive : 표준입력이나 표준출력대신 주어진 파일이름으로 압축한다.
--force-local : -F, -I, -O추가선택과 함께 압축된 파일을 국부파일로 인식한다.
-H FORMAT, --format=FORMAT : 주어진 FORMAT형식으로 압축한다.

bin	2진형식
odc	이전의 POSOX이식형식
newc	새로운 SVR4이식형식
crc	새로운 SVR4이식형식에 검사합추가
tar	tar형식
ustar	POSIX.1 tar형식
hpbin	HPUX의 cpio에 쓰이는 2진형식
hpodc	HPUX의 cpio에 사용되는 이식형식

-i archive : 표준입력대신 지정한 파일이름을 쓴다.
-l, --link : 복사하는것이 아니라 파일을 련결한다.
-L, --dereference : 련결을 복사하는것이 아니라 원본파일을 복사한다.
-m, --preserve-modification-time : 파일을 만들 때 파일수정회수도 함께 보관한다.
-M MESSAGE, --message=MESSAGE : 플로피디스크와 같이 매체여벌복사크기에 도달했을 때 지정한 MESSAGE를 보여준 후 입력재촉문을 내보내여 새 매체를 삽입하게 한다.
-n,--numeric-uid-gid : 실행시 동작하는 파일이름을 보여주는것이 아니라 UID와 GID를 보여준다.
-O archive : 표준출력대신 지정한 파일이름을 사용한다.
--quiet : 복사된 블로크수를 표시하지 않는다.
-r, --rename : 대화식으로 파일을 다시 지정한다.
-R [user][:.][group], --owner [user][:.][group] : -o과 -p추가선택에서 지정한 소유권으로 파일을 생성한다.
-t, --list : 입력목록표를 출력한다.
-u, --unconditional : 이전의 파일이 있든 없든 상관없이 무조건 파일을 복구한다.
-v, --verbose : 실행하는 동안 파일들을 보여주거나 -t추가선택과 함께 쓰이여 ls -l명령의 형식으로 목록을 보여준다.
--version : 판본정보를 보여준다.

파일을 복사하고 압축하는 지령으로서 체계의 여벌복사에 자주 사용된다.

crontab

crontab관리

지령형식: crontab [-u 사용자 ID] 파일

crontab [-u 사용자 ID] { -l | -r | -e }

-l : 현재 crontab내용을 표준출력으로 보여준다.
-r : 현재 crontab내용을 지운다.
-e : crontab파일의 내용을 편집한다. VISUAL이나 EDITOR환경변수의 편집기를 사용한다.

crontab는 지정한 시간에 지정한 작업을 하도록 설정하기 위한 주기적인 작업일정 작성지령이다. 이와는 달리 at지령은 지정한 시간에 작업이 한번만 진행된다.

crontab설정은 /etc/crontab에서 할수 있다.

그러면 root계정에 새로운 작업일정을 등록해보자.
먼저 이전에 crontab에 설정된 작업이 있는가를 확인한다.

우에서 보는바와 같이 등록된 작업이 없다.
-e추가선택으로 편집을 하게 되면 편집기가 실행되여 새로운 cron작업을 등록할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# crontab -e
no crontab for root - using an empty one
```

이 상태에서 다음과 같이 입력해보자.

```
0 4 * * * echo crontab testing
```

보관하고 탈퇴하면 아래의 통보문을 볼수 있을것이다.

```
"crontab.3697" 1L, 31C written
crontab: installing new crontab
```

그러면 등록된 작업을 -l추가선택으로 살펴보자.

```
root@Dolphin:~
[root@Dolphin root]# crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.3697 installed on Mon Feb  7 17:52:38 2005)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
0 4 * * * echo crontab testing
```

cron일정작성에 등록하는 시간지정에 대한 매 항목은 다음과 같다.

nutes분(M)	0-59
시(Hur)	0-23
일(Day of month)	1-31
월(Month)	1-12 혹은 Jan, Feb, Mar, ...
요일(Day of week)	0-6(0은 Sunday) 혹은 Sun, Mon, Tue, ...

웨브봉사기의 종류

현재 Internet상에서 사용하고있는 웨브봉사기에는 여러가지가 있지만 제일 많이 사용하는 대표적인 웨브봉사기로서는 apache를 들수 있으며 Microsoft회사의 IIS도 많이 사용되고있다.

전체 웨브봉사기의 60%정도가 apache이며 IIS는 30%정도이다.

Csplit

/usr/bin/csplit

파일분할

지령형식: cspilt [추가선택]... 파일 인자 ...

- : 표준입력으로부터 입력을 받는다,

-b suffix, --suffix-format=suffix : 새롭게 생성되는 파일의 뒤붙이를 지정한다. 보통 파일은 00, 01로 시작해서 99까지 생성된다. 아래와 같은 형식지정이 가능하다.

- %d 부호있는 10진수
- %i %d와 같다.
- %u 부호없는 10진수
- %o 8진수
- %x 16진수
- %X %x와 같다.

-f prefix, --prefix=prefix : 새롭게 생성되는 파일의 앞붙이를 지정할수 있다. 다시말하여 prefix00에서 prefix99까지 prefix부분을 지정하는것이다,

-k, --keep-files : 파일생성중에 오유가 발생하여도 새롭게 생성된 파일들을 유지한다.

-n, --digits=num : 생성되는 파일의 길이를 지정한다. 기정값은 두자리수이다.

-s, --quiet, --silent : 생성되는 파일의 크기를 보여주지 않는다.

-z, --elide-empty-files : 빈 파일을 제거한다.

--help : 도움말을 보여준다.

--version : 판본정보를 보여준다.

지정한 기준으로 파일을 자동으로 나누어주는 지령으로서 내용이 긴 문서파일을 효과적으로 나누어 읽을수 있게 한다.

아래에는 파일이름 다음에 올수 있는 인자(파일들을 나누는 기준)들을 설명하였다.

- n : 현재행부터 n번째 행까지의 범위를 파일로 생성한다.
- 정규표현식/[+n/-n] : 현재행부터 정규표현식을 포함하는 행까지 파일을 생성한다. +n이면 정규표현식을 기준으로 n번째 행까지 더 포함하며 -n이면 정규표현식을 기준으로 n번째 행까지를 제외하게 된다.
- %정규표현식%/[+n/-n] : 정규표현식의 앞부분은 보관되지 않는다.
- {n} : n번 반복한다.
- {*} : 입력이 있을 때까지 반복한다.

실례로 영어본문을 보관한 text.txt파일을 3개의 파일로 나누어 보관해보자.

```
[root@Dolphin root]# cat text.txt
PART 1
Hello, Mr Kim.
How are you this morning.
Fine thanks. And you?
I'm fine too.
Good bye.
Good bye.

PART 2
Hey, Comrade Pak. This is my syster, Ms Yong Hui.
How do you do.
How do you do. I'm glad to meet you.
See you again.
Goo bye.

PART 3
This is my room.
I get up at 6:30.
I clean my face, have breakfast at 7 o'clock.
I go to school at 7:30
School is over at 12.
You have new mail in /var/spool/mail/root
```

이 파일을 매 PART별로 나누어보자.

```
[root@Dolphin root]# csplit text.txt %PART1% /PART2/ /PART3/
104
133
132
```

csplit지령으로 xx00, xx01, xx02라는 파일들이 생성되였다. 새로 생성되는 파일의 번호가 00부터 시작한다는데 주의하여야 한다.

cut

/bin/cut

마당추출

지령형식: cut [추가선택]... [파일]...

-b, --bytes=LIST : byte단위로 나타낸다
-c, --characters=LIST : 문자단위로 나타낸다.
-d, --delimiter=DELIM : 지정된 마당구분자를 사용한다. 기정의 마당구분자는 TAB이다.
-f, --fields=LIST : 지정한 마당만을 보여준다.
-s, --only-delimited : 마당구분자를 포함하지 않는 행은 보여주지 않는다.
--output-delimiter=STRING : 출력되는 마당구분자를 지정된것으로 사용한다.
--help : 도움말정보를 보여준다.
--version : 판본정보를 보여준다.

cut지령은 파일에서 원하는 마당을 추출하는 지령이다. awk에서 print $N과 비슷하지만 awk보다 더 간단하게 사용될수 있다. -b, -c, -f추가선택에서는 다음과 같이 일정한 수자범위를 쓸수 있다.

- N : N번째
- N- : N번째부터 마지막까지의 범위
- N-M : N번째부터 M번째까지의 범위
- -N : 첫번째부터 N번째까지의 범위

OS와 핵심부판본 얻기

장비된 파일체계의 목록 얻기

date

/bin/date

날자보기

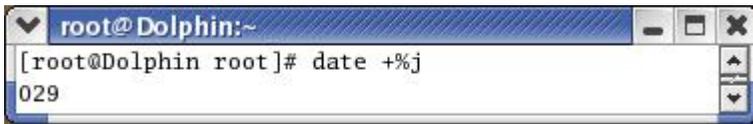
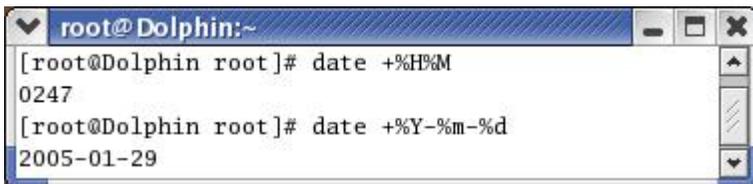
지령형식: date [추가선택] [+형식] [날자]

-d, --date=string : string에 설정한 값에 해당한 시간과 날자를 보여준다.
-f, --file=datedate : 날자정보를 보관하고있는 파일에서 매 행에 대한 시간과 날자를 보여준다.
-I, --iso-8601[=timespec] : ISO-8601형식으로 보여준다. timespec를 지정해주면 날자나 시,

분, 초에 대하여 해당한 값을 보여준다.
-r, --reference=file : 지정한 파일이 마지막으로 수정된 날자를 보여준다.
-R, --rfc-822 : RFC-822방식으로 날자를 보여준다.
-s, --set=string : string에 설정된 값으로 시간을 맞춘다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

date지령은 시간과 날자를 다양한 형식으로 출력하는 지령이다. 만일 추가선택이 없으면 체계의 현재의 시간과 날자를 보여주며 +로 시작되는 추가선택이 있으면 지시하는 형태대로 보여주게 된다.

기본출력은 다음과 같다.

%j는 현재 체계에 설정된 년도에서 오늘 날자까지 며칠이 지났는가를 나타낸다.

%H는 현재의 시간을 24시간제로 보여주며 %M은 분을 보여준다.
또한 다음과 같은 방법으로 오늘 날자의 년월일을 형식을 지정하여 표현할수도 있다.

--date추가선택으로 현재시간에서 15일후의 날자를 알수 있다.

그러면 현재 시간에서 3년 2개월전의 날자를 표시해보자.

이와 같은 날자의 출력결과를 다양한 형식으로 할수 있다. 또한 쉘스크립트와 cron을 리용하여 여벌복사나 기타 작업을 할 때 쉘스크립트에서 date지령을 리용하여 작업한 날자와 시간으로 파일이름을 만들도록 지정할수도 있다.

%	%문자 그자체
-	마당을 채우지 않는다. (기정으로 0을 채운다.)
_	마당을 공백으로 채운다.
%a	간략된 요일이름 (Sun...Sat)
%b	간략된 월이름 (Jan...Dec)
%c	날자와 시간(Mon Jun 01 10:42:09 EST 2002)
%d	월 기준일(01...31)
%h	%b와 같다.
%j	년 기준일(001...366)
%k	24시간 기준시간(0...23)
%l	12시간 기준시간(1...12)
%m	년 기준월(01...12)
%n	새로운 행을 입력한다.
%P	AM 혹은 PM
%r	12시간 표현형식(ff:mm:ss [AM | PM])
%S	1970-01-01 00:00:00 UTC로부터 경과된 초
%t	태브를 입력한다.
%W	요일의 수자(일요일은 0)
%x	날자표현식(mm/dd/yy)
%y	년표시에서 뒤의 두자리만 리용(00-99)
%z	RFC822 표현의 수자시간대
%A	완전한 요일표시 (Sunday...Saturday)
%B	완전한 월표시 (January...December)
%D	%m/%d/%y날자표현식
%H	24시간 표현형식 (00...23)
%l	12시간 표현방식 (01..12)

%M	분표시 (00...59)
%S	초표시 (00...59)
%T	%H:%M:%S형태의 시간표시
%U	몇번째 주인가를 나타내는 수자 (일요일을 기준 00...53)
%V	몇번째 주인가를 나타내는 수자 (월요일을 기준 01...53)
%W	몇번째 주인가를 나타내는 수자 (월요일을 기준 00...53)
%X	정의한 시간의 표시 (%H:%M:%S)
%Y	네 자리수자로 된 년표시 (례; 2002)
%Z	시간대의 이름 (례; EST)

dd

/bin/dd

블로크단위로 복사

지령형식: dd [추가선택]

--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.
of=file : 표준출력으로가 아니라 지정한 파일에 출력한다.
if=file : 표준입력으로가 아니라 지정한 파일에서 읽어들인다.
ibs=BYTES : 한번에 지정한 byte수만큼 읽는다.
obs= BYTES : 한번에 지정한 byte수만큼 쓴다
bs= BYTES : 지정한 byte수만큼 읽고 쓴다.(ibs, obs값은 무시된다.)
count=BLOCKS : 입력블로크를 지정한 블로크수만큼 복사한다.
conv=ascii : EBCDIC코드를 ASCII코드로 변환한다.
conv=ebcdic : ASCII코드를 EBCDIC코드로 변환한다.
conv=ibm : ASCII코드를 호환 EBCDIC코드로 변환한다.
conv=swab : 입력되는 두 바이트의 순서를 바꾼다. 입력파일에서 짝이 맞지 않는 하나의 바이트가 남으면 마지막 바이트는 그냥 복사된다.

dd지령은 기동디스크나 교환파일을 만드는 등의 특수한 복사를 진행하는 지령이다.

ASCII-EBCDIC사이의 변환, 대소문자변환은 물론 입출력바이트쌍을 바꾸거나 입력파일의 처음이나 끝을 잘라내서 출력파일을 만드는 등의 다양한 변환을 할수 있다. 여벌복사에서처럼 많은 자료를 복사할 때도 사용한다. dd는 파일내용을 장치에 그대로 쓴다.

주의 Windows에서 기동디스크를 작성할 때는 rawrite를 사용한다.

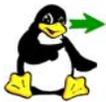
기동디스크 만들기

kernel-image는 /boot등록부에 있는 핵심부사본파일의 이름이다.

512MB의 교환파일 만들기

```
# dd if=/dev/zero of=/swap bs=1024 count=524288
# mkswap/swap 524288
# sync
# swapon/swap
```

free지령으로 작성된 교환파일을 확인한다.

```
# free
```

련관지령

tar　빠르고 간단한 파일묶기지령
cpio　체계여벌복사에 사용하는 지령
mt　자기테프조작지령

lilo에 통과어 설정

Linux가 설치된 콤퓨터를 보호하기 위하여 기동적재기인 lilo에 통과어를 설정해보자.
/etc/lilo.conf파일을 열어 원하는 위치아래에 다음과 같은 내용을 추가한다.

```
password=원하는 통과어
```

이때 password만 지정하면 어떤 기동방법을 사용하더라도 통과어를 묻는다. 즉 init 3이나 init 5로 기동할 경우에도 통과어를 묻게 되므로 원격으로 기동할 때에는 문제가 생길수 있다. 이것을 방지하려면 통과어를 설정한 다음행에 restricted라고 입력하고 보관한다.

debugfs

/sbin/debugfs

파일체계조종

지령형식: debugfs [[추가선택] 장치이름]

-w : 읽기/쓰기방식으로 파일체계를 연다.

debugfs지령은 ext2파일체계의 오유제거기로서 파일체계를 직접 확인하며 조종할수 있다. debugfs지령을 실행한 다음 아래와 같은 지령들을 사용할수 있다.

cat 파일이름	inode 하나의 내용을 표준출력으로 내보낸다.
cd 등록부	작업등록부를 변경한다.
chroot 등록부	지정한 inode로 root등록부를 변화시킨다.
close	열려진 파일체계를 닫는다.
clri 파일이름	파일에 해당하는 inode의 내용을 지운다.
dump <inode> 출력파일이름	inode의 내용을 파일로 옮긴다.
expanddir 파일이름	등록부를 확장한다.
find_free_block [목표]	목표로부터 시작해서 첫번째 빈 블로크를 찾아 할당한다.
find_free_inode [등록부[모드]]	빈 inode를 찾아서 할당한다.
freeb 블로크	블로크를 할당되지 않은것으로 표시한다.
freei 파일이름	파일이름에 해당하는 inode를 해제한다.

help	debugfs가 사용할수 있는 지령목록을 보여준다.
check 블로크	블로크를 inode로 변환한다.
iname node	inode에 해당한 파일이름을 보여준다.
initialize 장치이름 블로크크기	지정한 장치에 ext2파일체계를 생성한다.
kill_file 파일이름	파일을 삭제하고 사용한 블로크를 해제한다.
ln 원본파일 목표파일	련결을 만든다.
ls [경로이름]	ls지령을 모방한다.
modify_inode 파일이름	파일이름에 해당하는 inode의 내용을 수정한다.
mkdir등록부이름	지정한 등록부이름으로 등록부를 생성한다.
mknod 파일이름 [pl[[clb] major번호 minor번호]]	특별한 장치파일을 지정한 major번호와 minor 번호로 생성한다.
open [-w] 장치이름	파일체계를 연다.
pwd	현재 작업등록부를 보여준다.
quit	debugfs를 완료한다.
rm 파일이름	지정한 파일을 지운다.
rmdir	지정한 등록부를 삭제한다.
setb블로크	지정한 블로크를 할당된것으로 표시한다.
seti 파일이름	파일이름에 해당하는 inode를 사용하는것으로 표시한다.
show_super_stats	슈퍼블로크의 목록을 보여준다.
stat 파일이름	지정한 파일이름으로 inode의 목록을 내보낸다.
estb 블로크	블로크가 할당되여있는가를 검사한다.
testi 파일이름	파일이름에 해당하는 inode할당이 되여있는가를 검사한다.
unlink 파일	지정한 파일의 련결을 제거한다.
write 원본파일 목적파일	파일이름을 가진 파일체계에 파일을 하나 만들고 원본파일의 내용을 목적파일에 복사한다.

주의 이 지령은 파일이 저장된 파일체계를 직접 조종하므로 블로크 등을 잘못 삭제할 경우 해당한 파일을 사용할수 없게 될수도 있다.

rm지령으로 삭제된 파일 복구하기

Linux에서는 삭제된 파일의 복구가 거의 불가능하다. 그러나 debugfs지령으로 방금 삭제된 파일을 한번 복구해보자.

<u>경고:</u> 파일을 삭제한 다음 새로운 파일을 보관하거나 프로그람을 설치하면 원래파일이 사용하던 블로크우에 덧쓰일수 있다. 이렇게 덧쓰기된 경우에는 파일을 복구할수 없다.

파일을 삭제할 등록부로 이동한다.

```
# cd /root
```

root등록부에 있는 2.txt파일을 삭제한다.

```
# rm -rf 2.txt
```

파일이 있는 구획에서 debugfs지령으로 오유제거를 진행한다.

```
root@Dolphin:~
[root@Dolphin dir]# debugfs /dev/hda6
debugfs 1.32 (09-Nov-2002)
debugfs:  lsdel
 Inode  Owner  Mode    Size    Blocks   Time deleted
725762      0 100600      64    1/    1 Sun Feb 13 01:03:20 2005
 22501      0 100600     212    1/    1 Sun Feb 13 01:03:21 2005
 22650      0 100600     212    1/    1 Sun Feb 13 01:03:21 2005
609329      4 100600     571    1/    1 Thu Feb 17 15:42:55 2005
609330      4 100600     690    1/    1 Thu Feb 17 15:42:55 2005
333272      0 100777 417792   44/  103 Fri Feb 18 12:26:33 2005
333275      0 100777 839680  203/  206 Fri Feb 18 12:26:34 2005
........................
379196      0 100600      85    1/    1 Mon Feb 21 18:51:20 2005
267815      0 100644   38075   10/   10 Mon Feb 21 18:51:35 2005
709407      0 100644     526    1/    1 Mon Feb 21 18:51:53 2005
102 deleted inodes found.
debugfs:
```

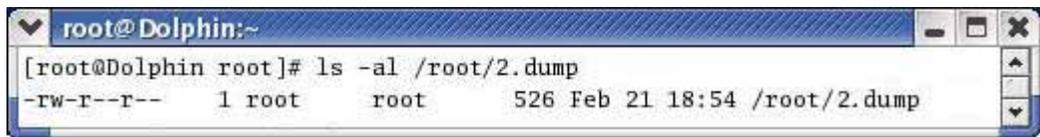
lsdel지령으로 삭제된 파일을 살펴볼수 있다.

709407의 inode값을 가진 파일이 2005년 2월 21일 18시 51분쯤에 삭제된것을 볼수 있다.

```
debugfs: dump <709407> /root/2.dump
```

dump지령으로 709407파일을 다른 파일체계에 복구한다.

```
debugfs: quit
```

이전에 삭제되였던 2.txt파일이 2.dump파일로 복구되였다.

depmod

/sbin/depmod

모듈의존성확인

지령형식: depmod [추가선택] 모듈이름

-a : /etc/modules.conf에 있는 모든 모듈의 의존성을 생성한다.
-d : 오유제거방식.
-e : 해결되지 않는 모듈의 목록을 보여준다.
-v : 처리된 모듈의 목록을 보여준다.

핵심부모듈의 의존성관계를 확인하는 지령이다.

이와 관련된 파일은 /etc/modules.conf이다.(Red Hat 7.0 아래 판본이나 다른 배포판에서는 /etc/conf.modules파일일수 있다.)

사용법은 modprobe지령과 같이 설명하였으므로 그것을 참고하기 바란다.

련관지령

insmod 모듈적재지령
modprobe 보다 우월한 모듈적재지령

df

/bin/df

구획에서 남은 용량 보기

지령형식: df [추가선택] [파일이름...]

-a, --all : 0블로크의 파일체계를 포함한 모든 파일체계를 보여준다.
-i, --inodes : inode의 남은 공간, 사용공간, 퍼센트정보를 보여준다.
-k, --kilobytes : 1KB단위로 보여준다.
-P, --portability : POSIX에서 사용되는 형태로 보여준다.
-T, --print-type : 매 구획의 정보에 그 파일체계의 형태까지 함께 보여준다.
-t type : 지정한 파일체계형태의 정보만을 보여준다.
-X type : 지정한 파일체계형태를 제외한 나머지 모든 파일체계정보를 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

df지령은 현재 탑재되여있는 파일체계의 사용정보를 보여주는 지령으로서 1024byte의 블로크단위로 보여준다.

주의 파일체계의 총 용량의 5%이하가 남았을 때 일반사용자는 체계에 등록가입할수 없다. 분담몫을 사용하거나 일정한 인원별로 구획을 나누어놓는 방법을 사용할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda6              6049740   5531012    211416  97% /
/dev/hda5             25170016  20516864   4653152  82% /WinData
/dev/hda1              7160404   4937764   2222640  69% /WinSys
none                    123592         0    123592   0% /dev/shm
```

inode의 사용정보를 살펴보자. 이 inode가 부족하다면 사용공간이 남아있더라도 더 이상 파일을 새롭게 만들수 없기때문에 살펴볼 필요가 있다.

```
root@Dolphin:~
[root@Dolphin root]# df -i
Filesystem            Inodes   IUsed   IFree IUse% Mounted on
/dev/hda6             768544  294553  473991   39% /
/dev/hda5                  0       0       0    -  /WinData
/dev/hda1                  0       0       0    -  /WinSys
none                   30898       1   30897    1% /dev/shm
```

-h추가선택은 KB단위가 아닌 GB, MB의 단위로 파일체계의 용량을 표시한다.

련관지령

du　디스크의 사용상태를 볼수 있는 지령

free　현재 기억기의 사용상태를 볼수 있는 지령

diff

/usr/bin/diff

파일비교

지령형식: diff [추가선택] 기준파일 비교파일

-a, --text : 모든 파일을 본문파일로 취급한다.
-b, --ignore-all-space : 중복된 공백과 행의 끝에 있는 공백을 무시한다.
-B, --ignore-blank-lines : 파일에 있는 공백을 무시한다.
-C, -C num, --context[=num] : 문맥상으로 바뀐 행을 지정한 수만큼 보여준다. (기정값은 2)
-d, --minimal : 바뀐것을 자세하게 보여준다.
-D name, --ifdef=name : C원천파일안에서의 #ifdef에 있는 내용도 보여준다.
-e, --ed : ed스크립트를 리용하여 보여준다.
-F regexp, --show-function-linux=regexp : 지정한 정규표현식에 맞는 행부터 보여준다.
-H, --speed-large-files : 큰 파일을 빨리 처리하기 위해 사용된다.
--help : 간단한 도움말을 보여준다.
-i, --ignore-case : 대소문자를 구별하지 않는다
-I regexp, --ignore-matching-lines=regexp : 지정한 정규표현식형태로 바뀐 부분은 무시한다.
-L label, --label=label : 파일이름대신에 지정한 표식을 사용한다.
-n, --rcs : RCS diff형식으로 보여준다.
-N , --new-file : 존재하지 않는 파일을 빈 파일로 인식한다.
-p, --unidirectional-new-file : 등록부를 비교할 때 두개의 등록부중 두번째 등록부에만 파일이 존재한다면 첫번째 등록부에는 사실 파일이 없지만 있는것처럼 처리한다.
-q, --brief : 파일이 차이날 때에만 보여준다.
-r, --recusive : 비교대상등록부의 하위등록부나 파일까지도 비교한다.

-s, --report-identical-files : 두 파일이 같은 경우에 보여준다.
-S file, --starting-file=filename : 등록부를 비교할 때 지정한 파일이름을 우선적으로 목록에서 보여준다.
-t, --expand-tabs : 태브로 정렬시키기 위해 출력시 태브의 공간을 입력파일에서 확장한다.
-T, --initial-tab : 들여쓰기를 위해 처음에 태브를 삽입한다.
-u, -U n, —unified[=n] : 문맥상으로 합쳐진 행을 지정한 수만큼 보여준다. (기정값은 2이다.)
-v, --version : 판본정보를 보여준다.
-w, --ignore-all-space : 두 행을 비교할 때 공백을 무시한다.
-W n, --width=n : 한 행의 최대문자수를 n으로 지정한다. (기정값은 130이다)
-x regexp, --exclude=regexp : 지정한 정규표현식과 같은 파일은 제외한다.
-x filename, --exclude-from=filename : 지정한 파일이름과 일치하는 파일과 등록부는 비교하지 않는다.
-y, --side-by-side : 비교결과로써 두개의 렬을 보여준다.

비교되는 두 파일의 차이나는 부분을 실제 두 파일의 내용과 함께 보여주는 지령이다. 실례를 들어보자. 아래의 실행화면은 hello.c와 hello1.c를 분석한것이다.

우의 실행화면에서 "4,5c4,5"는 앞파일에서는 4, 5행이, 뒤파일에서는 4, 5행이 다르다는것을 나타낸다. ">"표시는 앞파일을 나타내며 "<"표시는 뒤파일의 내용을 나타낸다.

주의 세 파일을 서로 비교할 때에는 diff3지령을 리용한다.

이런 방법을 사용하여 어떤 원천의 수정된 내용을 얻었을 때 수정된 내용들사이에서 어디가 어떻게 달라졌는가를 쉽게 확인할수 있다
또한 프로그람을 설치할 때마다 어떤 프로그람이 추가되였는가를 쉽게 확인하려 할 때 프로그람을 설치한 다음 find지령으로서 전체 파일의 구체적인 내용을 파일로 보관하고 새 프로그람을 설치할 때마다 diff지령을 사용하여 확인할수도 있다.

수정파일 만들기 용어상식

원천코드를 약간 수정하여 달라진 부분을 손쉽게 적용할수 있도록 만든 파일을 수정(patch)파일이라고 한다. 수정파일은 diff지령을 사용하여 손쉽게 만들수 있다.

song2.3이라는 프로그람의 원천코드가 song2.3이라는 등록부안에 들어있다고 하자. 원래의 원천코드등록부이름을 song2.3-orig로 변경하고 새로 추가한 원천코드등록부이름을 song2.3으로 한다. 그리고 다음과 같은 명령을 실행하면 song2.3-new.diff라는 수정파일을 만들수 있다.

```
# diff -urN song2.3-orig song2.3 > song2.3-new.diff
```

이 diff파일안에는 두 등록부의 원천코드에서 변경된 부분이 보관되여있다. ---로 시작하는 행은 원본파일, +++로 시작하는 행은 수정파일의 내용이며 -표시가 붙은 행은 삭제하고 +표시가 붙은 행은 추가하거나 바꾸어넣는다는 뜻이다.

dirname

/usr/bin/dirname

경로이름보기

지령형식: dirname 파일이름

--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

전체 경로에서 파일이름을 제외하고 등록부의 이름만을 보여주는 지령으로서 지정한 파일이나 등록부에 대한 등록부구조에서 마지막 준위를 제외한 등록부의 경로를 보여준다. 만일 전체 절대경로를 지정하지 않으면 현재 등록부를 표시하는 "."을 보여주게 된다.

절대경로와 상대경로

콤퓨터조작체계에서 경로(path)란 파일체계에서 지정한 파일까지의 등록부구조를 말한다. 경로이름(pathname)이란 경로를 자세히 렬거한것이다. 매 조작체계는 경로이름을 지정하기 위한 고유한 형식을 가지고있다.

Linux계렬의 체계들에서 경로의 형식은 다음과 같다.

```
/directory/subdirectory/filename
```

절대경로이름이란 뿌리등록부로부터 시작하여 완전히 서술한 경로이름을 말한다. 상대경로이름은 현재 설정해놓은 등록부로부터 시작하여 상대적인 경로를 지정한다.

dmesg

/bin/dmesg

기동정보보기

지령형식: dmesg [추가선택]

-c : 통보문내용을 보여주고 순환완충기를 모두 비운다.
-n level : 통보문이 조작탁에서 표시되는 준위를 설정한다.

dmegs지령은 체계를 기동할 때 적재되는 체계정보들의 각종 통보문을 볼 때 리용한다. 이 정보는 Linux핵심부판본과 gcc판본, CPU정보 등 기동시의 체계정보들을 포함하고있다.

화면으로부터 이 체계는 Linux핵심부 2.4.20을 사용하고 gcc판본은 3.2.2, 기억기는 248MB정도이며 CPU는 Intel Pentium(Intel(R) Celeron(R) 1.8GHz)이라는것을 알수 있다.

핵심부판본의 의미

먼저 현재 최신의 핵심부판본정보를 확인해보자.

```
# finger @ finger kernel.org
```

실행결과로부터 최신의 안정판본과 개발판본정보를 확인할수 있다. 여기에서 2.4.20 핵심부판본은 다음과 같이 세부분으로 나누어 생각할수 있다.

- 2　몇번 수정되였는가를 나타낸다.
- 4　안정판본인가 또는 개발판본인가를 나타낸다.(홀수:개발, 짝수:안정)
- 20　부분적인 변화가 있을 때 바뀐다.

이 의미대로 생각해보면 2.4대의 안정판본으로서 수정이 20번 진행되여 추가되였다는것이다. 개발판본인 경우 충분한 수정과 실험이 이루어져 안정성이 검증되면 짝수판본으로 넘어간다. 핵심부개발자가 아닌 사용자는 안정판본정보를 사용하는것이 좋다.

dnsdomainname

/bin/dnsdomainname

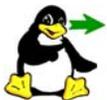
DNS정보확인

지령형식: dnsdomainname [-v]

체계의 DNS봉사기이름과 정보를 볼수 있는 지령이다.

현재 체계의 DNS봉사기를 알아보자.

-v추가선택은 좀 더 자세한 정보를 보여준다.

련관지령

hostname	체계의 콤퓨터이름을 설정하거나 보여주는 지령
domainname	체계의 NIS/YP령역이름을 설정하거나 보여주는 지령
nisdomainname	체계의 NIS/YP령역이름을 설정하거나 보여주는 지령
ypdomainname	체계의 NIS/YP령역이름을 설정하거나 보여주는 지령

domainname

/bin/domainname

NIS령역보기

지령형식: domainname [추가선택] [이름]

-s, --short : 간략된 콤퓨터이름을 보여준다.
-a, --alias : 콤퓨터의 별명을 보여준다.
-i, --ip-address : 콤퓨터의 IP주소를 보여준다.
-f, --fqdn, --long : 콤퓨터이름을 보여준다.
-d, --domain : DNS령역이름을 보여준다.
-y, --yp, --nis : NIS/YP령역이름을 보여준다.
-F, --file : 콤퓨터이름이나 NIS령역이름을 지정된 파일에서 읽어 보여준다.

domainname지령은 NIS/YP지령으로서 현재의 NIS령역이나 설정된 내용을 보여주는 지령이다. 령역이름을 변경해본다.

련관지령

hostname	체계의 콤퓨터이름을 설정하거나 보여주는 지령
dnsdomainname	체계의 DNS봉사기이름과 정보를 보여주는 지령
nisdomainname	체계의 NIS/YP령역이름을 설정하거나 보여주는 지령
ypdomainname	체계의 NIS/YP령역이름을 설정하거나 보여주는 지령

NIS/YP

NIS는 sun회사에서 개발한 망봉사규약으로서 Network Information service의 략자이다. 원래 YP(Yellow Pages)라는 이름으로 부르기도 했지만 Yellow Pages라는 이름이 이미 등록되여있어 허가없이는 사용할수 없었기때문에 이후에 NIS라는 이름으로 등록하였다. NIS는 Sun Microsystems회사의 등록상표이다.

NIS는 망에서 다른 콤퓨터에 정보를 전달하는 방식으로 망상에 있는 모든 체계에 봉사기의 /etc/passwd파일에 저장된 등록가입이름과 통과어, home등록부의 정보와 /etc/group에 저장된 그룹정보를 제공한다. NIS주봉사기는 정보표를 가지고있으며 그것들을 NIS대응(map)파일들로 변환한다. 이 대응파일들이 망을 통해 제공됨으로써 NIS의뢰기들은 그와 같은 여러가지 정보를 얻을수 있다.

NIS를 리용하면 NIS자료기지에 등록된 사용자들은 이 망에서 NIS의뢰기가 설치되여있는 어떤 체계에도 등록가입할수 있으며 통과어를 한번만 바꾸면 그 NIS령역에 들어있는 모든 콤퓨터의 등록가입정보를 갱신할수 있다.

사실 NIS는 안전한것이 아니다. 안전성보다 사용상 편리를 위하여 만든것이므로 NIS를 사용할 때에는 보안상 주의해야 한다. 봉사기의 NIS령역이름을 알아내는 사람은 여러 방법으로 passwd파일을 복사할수 있으며 사용자통과어를 없애기 위해 여러가지 통과어파괴도구를 사용할수도 있다.

이런 점을 조금이나마 방지하자면 NIS+를 사용해야 한다. NIS+에 대해서는 관련참고서들을 참고하기 바란다.

dosfsck

/sbin/dosfsck

MS-DOS파일체계 복구

지령형식: dosfsck [추가선택...] 장치이름

-a : 파일체계를 자동적으로 복구한다.
-d file : 파일배치표에서 지정한 파일을 지운다.
-f : 파괴된 련결을 파일로 복구한다.
-l : 경로이름을 보여준다.
-r : 파일체계를 복구할 때 일일이 물어본다.

-t : 불량클라스터에 대한 검사를 진행한다.
-u file : 지정한 파일을 삭제하지 않는다.
-v : 보다 상세하게 보여준다.
-V : 확인절차를 생략한다.
-w : 변경된 내용을 즉시 디스크에 반영한다.
-y : -a추가선택과 같다.

dosfsck지령은 MS-DOS파일체계를 검사하고 복구하는 체계관리지령으로서 fsck와 류사하다. 하지만 이것은 MS-DOS의 파일체계에만 적용할수 있다. Linux는 MS-DOS, Sun 등의 다양한 환경에서 파일체계를 읽기/쓰기하거나 읽을수 있게 한다. 이것은 다양한 조작체계에서 사용하는 파일체계를 지원하기 위해서이다. 그러나 파일체계마다 고유한 특징이 있기때문에 파일체계를 검사하고 복구하자면 서로 다른 지령이 필요하다. Linux에서 다른 조작체계에 대한 파일체계를 검사하고 복구하는 지령가운데서 대표적인것은 이 FAT파일체계에 대한것과 MINIX파일체계에 대한것이다.

du

/usr/bin/du

자체 사용용량 보기

지령형식: du [추가선택...] [파일...]

-a, --all : 현재 등록부에 들어있는 모든 파일과 등록부의 사용정보를 보여준다.
-B, --block-size=size : 지정한 size를 블로크크기로 사용한다.
-b, --bytes : byte크기로 보여준다.
-c, --total : 모든 파일의 디스크사용정보를 보여준 다음 합계를 보여준다. 이것을 리용하면 해당한 경로가 디스크공간을 얼마나 사용하고있는가를 알수 있다.
-D, --dereference-args : 계산되는 파일이나 경로가 기호련결파일이면 그 원본의 값을 보여준다.
-h, --human-readable : 파일용량을 읽기 쉬운 형태로 보여준다. 실례로 1KB, 234MB, 2GB 등
-H, --si : -h와 비슷하나 1024단위의 비률로 보여준다.
-k : 단위를 1KB형태로 보여준다.
-l --count-links : 계산되는 파일중에 하드련결되여있는 파일이 있다면 용량을 그대로 계산한다.
-L --dereference : 모든 기호련결을 따른다.
-S --separate-dirs : 등록부의 총 사용량을 보여줄 때 하위등록부의 사용량은 제외한다.
-s, --summarize : 총 사용량만 간단히 보여준다.
-x, --one-file-system : 현재 파일체계의 파일사용량만을 보여준다.
-x file, --exclude-from=file : 지정한 파일과 일치하는것은 제외한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

du지령은 현재등록부의 사용량을 보여준다. 기본단위는 MB이다. 현재경로의 모든 등록부의 크기를 나타내는데 -a추가선택을 리용하면 모든 파일들을 보여줄수도 있고 지정한 파일 또는 등록부를 요약하여 나타낼수도 있다.

-s추가선택은 지정한 파일이나 등록부의 총 사용량만을 보여준다.

-c는 현재등록부의 총 사용량을 보여주며 -h는 알아보기 쉽게 적당한 단위로 보여준다.

dumpe2fs

/sbin/dumpe2fs

파일체계정보 보기

지령형식: dumpe2fs 장치이름

dumpe2fs지령은 장치이름에 존재하는 ext2파일체계의 정보를 보여준다.

Linux파일체계

Linux파일체계는 5개의 주요부분으로 이루어져있다. 파일체계의 앞부분에는 기동블로크가 저장된다. 나머지 부분은 실린더그룹이라고 하며 여러 실린더그룹으로 나뉘여 관리된다. 하나의 실린더그룹은 슈퍼블로크, 실린더그룹블로크, inode표와 자료블로크로 이루어져있다.

(1) **기동블로크** : 기동에 필요한 파일들이 존재하며 root령역외에는 해당되지 않는다. 일반사용자에게는 해당되지 않는 블로크이다.

(2) **슈퍼블로크** : 파일체계의 크기, inode표의 크기, free블로크목록 등 파일체계를 관리하는데 필요한 정보가 기억된다.

이 부분에 오유가 생기면 이 실린더를 사용할수 없게 될수 있으므로 보통 슈퍼블로크는 여러 블로크에 여벌을 만들어놓는다.

(3) **실린더그룹블로크** : 실린더그룹에서 유효블로크들의 비트배치정보나 통계정보를 기록한다.

(4) **inode표** : 파일에 대한 중요한 정보를 가지고있는데 즉 파일크기, 자료위치(디스크주소), 파일류형, 사용허가권, 작성날자 등이 기록되며 핵심부의 파일관리에서 핵심으로 되는 모든 자료를 기록한다.

(5) **자료블로크** : 실제자료가 저장되는 공간이다.

```
root@Dolphin:/home/han
[root@Dolphin han]# dumpe2fs /dev/hda4
dumpe2fs 1.32 (09-Nov-2002)
Filesystem volume name:   /
Last mounted on:          <not available>
Filesystem UUID:          5d873186-a044-43d6-9f14-b9eb7c12a2c8
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      filetype sparse_super
Default mount options:    (none)
Filesystem state:         not clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              642560
Block count:              1283310
Reserved block count:     64165
Free blocks:              542321
Free inodes:              474750
First block:              0
Block size:               4096
Fragment size:            4096
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         16064
Inode blocks per group:   502
Filesystem created:       Wed Jan 26 09:43:14 2005
Last mount time:          Sat Jan 29 05:40:47 2005
Last write time:          Sat Jan 29 06:38:35 2005
Mount count:              3
Maximum mount count:      23
Last checked:             Sat Jan 29 05:25:25 2005
Check interval:           15552000 (6 months)
Next check after:         Thu Jul 28 06:25:25 2005
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
```

dumpkeys

/usr/bin/dumpkeys

건반의 코드값 보기

지령형식: dumpkeys [추가선택]

-i, --shot-info : 건반구동프로그람에 관한 정보를 보여준다.
-l, --long-info : -i추가선택에 추가적으로 핵심부에서 지원하는 건의 코드값과 자세한 정보를 보여준다.
-n, --numeric : 코드값을 16진수로 표시한다.
-f, --full-table : 간략형식이 아닌 완전한 형식의 표를 보여준다.
-l, --separate-lines : 매 modifier/keycode를 한행에 하나씩 보여주며 변경되지 않는 keycode의 앞에 자체문자를 붙여준다.
-S, --shape=num : 표형태를 결정하는 수값을 리용하여 정보를 보여준다. 매 수값은 다음과 같다.

0	기정값
1	--full-table과 같다.
2	--separate-lines와 같다.
3	keycode를 첫번째 행에 두고 modifier/keycode부분을 한행씩 보여준다.

-c, --charset=<charset-name> : 지정한 문자모임을 활성화코드로 해석한다.
사용할수 있는 문자모임은 다음과 같다.

iso-8859-1	iso-8859-2	iso-8859-3	iso-8859-4	iso-8859-5
iso-8859-7	iso-8859-8	iso-8859-9	iso-8859-14	iso-8859-15
mazovia	koi8-r	koi8-u	iso-10646-18	

-h, --help : 도움말을 보여준다.
-V, --version : 판본정보를 보여준다.

건반입력을 코드값으로 변환하는 지령이다. 자주 리용하는 지령은 아니다.

e2fsck

/sbin/e2fsck

Linux파일체계의 점검

지령형식: e2fsck [추가선택] 장치이름
fsck ext2 [추가선택] 장치이름

-b superblock : 기본슈퍼블로크를 사용하지 않고 지정한 슈퍼블로크를 불러온다.
-B blocksize : 슈퍼블로크를 찾기 위해 지정한 블로크크기를 사용한다.
-c : 불량블로크를 검사한다.
-f : 파일체계에 이상이 있든지 없든지 상관없이 파일체계를 검사한다.
-l bad_blocks_file : 지정한 불량블로크파일을 불량블로크목록에 추가한다.
-L bad_block_file : 지정한 불량블로크파일을 불량블로크목록으로 설정한다.
-n : 파일체계를 읽기전용으로 열고 질문에 대해 no로 한다.
-p : 사용자의견을 묻지 않고 자동으로 파일체계를 검사한다.
-v : 자세한 정보를 보여준다.
-y : 파일체계를 읽기전용으로 열고 질문에 대해 yes로 한다.

ext2파일체계를 점검하는 지령으로서 fsck지령에 -t ext2추가선택을 붙인것과 같다. fsck지령이나 e2fsck지령은 파일체계를 직접 검사하는 지령이므로 탑재를 해제한 상태에서 파일체계를 검사할것을 요구한다.

물론 일부러 파일체계를 검사할수도 있다. 그러나 체계에 이상이 생기면 기동할 때 대체로 e2fsck가 자동적으로 파일체계를 검사한다. 문제가 심각한 경우에는 파일체계를 수리하던중 다음과 같은 통보문을 보여주면서 중단하는 경우도 있다.

```
root@Dolphin:~
*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

만일 그대로 다시 기동하려면 Control+D를 누른다. root통과어를 입력한 경우에는 파일체계점검지령으로 체계를 검사할수 있다.

```
(Repair filesystem) # fsck /
```

주의 구획을 정확히 모르는 경우 fdisk -l명령으로 알아볼수 있다.

여기서 /는 Linux가 설치되여있는 분구이다. 현재 탑재되여있는 /dev/hda1 파일체계를 검사하려면 /hda1의 탑재를 해제한 다음 실행하여야 한다.

```
[root@Dolphin root]# umount /hdb6
[root@Dolphin root]# e2fsck /dev/hdb6
e2fsck 1.32 (09-Nov-2002)
/dev/hdb6 is mounted.

WARNING!!!  Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes

/dev/hdb6 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/hdb6: 102/1537088 files (0.0% non-contiguous), 51436/1536562 blocks
```

련관지령

fsck 파일체계를 점검하는 지령

echo

환경변수출력

지령형식: echo [-ne] [추가선택]

-n: 마지막에 행바꾸기(newline : 화면상에서 유표를 다음 행에 가도록 하는 코드로서 ASCII 문자모임에서 10진수로 10에 해당되는 값이다.)문자를 출력하지 않는다.
-e : 문자렬에서 역사선(\)과 조합되는 탈출문자렬(escape sequence : 인쇄기나 현시기의 화면속성 등을 조종하기 위해 장치에 보내는 특수문자렬을 의미하는데 그 중에서도 특히 탈출문자로 시작되는 기계지령을 말한다.)을 인용부호(")로 묶어서 인식하게 한다. 탈출문자렬에는 다음과 같은것들이 있다.

\a	경고음(삑)
\b	후퇴
\c	마지막에 행바꾸기문자를 사용하지 않는다.
\f	용지이송(form feed : 인쇄기에서 용지바꿈)
\n	행바꾸기문자
\r	복귀 (carriage return : 유표를 그 행의 맨 앞으로 옮기는데 사용되는 특수한 코드이다. ASCII문자모임에서 CR는 10진수 "13"에 해당되는 값이다.)
\t	수평태브
\v	수직태브
\\	역사선
\nnn	ASCII 코드가 nnn (8진수)인 문자

--help : 표준출력으로 도움말을 출력하고 정상적으로 완료한다.
--version : 표준출력으로 판본정보를 출력하고 정상적으로 완료한다.

체계의 환경변수 또는 입력내용을 화면에 출력한다.

echo지령은 뒤에 오는 문자렬을 출력하는 지령으로서 문자렬을 출력한 다음에는 행바꾸기문자를 덧붙여 출력한다. 또한 쉘자체에 포함하고있는 내부환경변수를 보는데도 리용할수 있다.

먼저 내부에서 사용하는 환경변수를 출력해보자.

```
root@Dolphin:~
[root@Dolphin root]# set | more
BASH=/bin/bash
BASH_ENV=/root/.bashrc
BASH_VERSINFO=([0]="2" [1]="05b" [2]="0" [3]="1" [4]="releas
e" [5]="i386-redhat-linux-gnu")
BASH_VERSION='2.05b.0(1)-release'
COLORS=/etc/DIR_COLORS.xterm
COLORTERM=gnome-terminal
COLUMNS=60
DESKTOP_STARTUP_ID=
DIRSTACK=()
DISPLAY=:0.0
EUID=0
GDMSESSION=Default
GNOME_DESKTOP_SESSION_ID=Default
GROUPS=()
GTK_RC_FILES=/etc/gtk/gtkrc:/root/.gtkrc-1.2-gnome2
G_BROKEN_FILENAMES=1
HISTFILE=/root/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/root
HOSTNAME=Dolphin
HOSTTYPE=i386
IFS=$' \t\n'
INPUTRC=/etc/inputrc
LANG=en_US.UTF-8
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=31
LOGNAME=root
--More--
```

이제 echo지령으로 환경변수 HOME을 출력하려면 환경변수이름앞에 $문자를 붙여야 한다.

환경변수를 보지 않고 자체 문자렬을 출력하려면 다음과 같다.

echo지령에서 표시되는 행바꾸기문자를 삭제해보자.

echo -e추가선택을 사용하여 탈출문자렬을 인식할수 있다. 체계에서 경고음을 울리자면 다음과 같이 할수 있다.

```
# echo -e "\a"
```

\t, \n, \r도 사용할수 있다.

eject

/usr/bin/eject

CD-ROM자동배출

지령형식: eject [추가선택] [장치]

-h, --help : 간단한 도움말을 보여준다.
-v, --verbose : 지령이 실행되는동안에 보다 구체적인 설명을 현시한다.
-d, --default : eject지령을 실행하는 기정장치의 이름을 보여준다.
-a, --auto on|off|1|0 : 자동배출방식을 on/off로 설정한다.
-c, --changerslot <slot> : ATAPI/IDE CD-ROM변환기에서 CD슬로트를 선택할수 있다. 현재 CD-ROM구동기가 작업중이면(실례로 자료CD가 탑재되였거나 음악CD가 재생되고있는 중) 이 지령을 쓸수 없다. 참고로 첫번째 슬로트는 1이 아니라 0이다.
-x, --cdspeed <speed> : CD-ROM속도를 정해주는 지령이다. 속도가 8배속이라면 8을 입력한다. 모든 장치에 대하여 다 적용하는것은 아니다.
-n, --noop : 실제 실행되는것이 아니라 배출되는 장치를 보여준다.
-r, --cdrom : CD-ROM을 배출한다.
-s, --scsi : scsi장치를 배출한다.
-f, --floppy : 플로피디스크를 배출한다.
-p, --proc : /etc/mtab대신 /proc/mounts파일을 리용한다.
-V, --version : 판본정보를 보여준다.

매체의 탑재상태를 해제한 다음 배출하는 지령이다. Linux설치가 끝난 후 CD가 자동적으로 튀어나오는것도 바로 이 지령을 사용하였기때문이다.

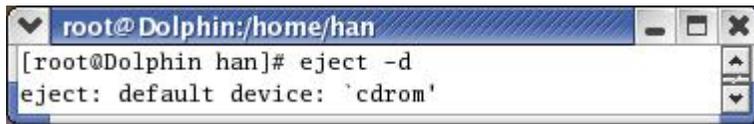
eject는 CD-ROM, 플로피디스크, 테프, JAZ나 ZIP디스크와 같은 장치를 배출할수 있다. 이 지령은 CD-ROM변환기나 자동배출을 지원하는 장치를 조종할수 있다.

다음의 지령을 리용하면 CD-ROM이 배출된다.

```
# eject
```

-t추가선택으로 CD-ROM을 넣을수도 있다.

```
# eject -t
```

-d추가선택으로 배출되는 장치이름을 볼수 있다.

SCSI (small computer system interface)

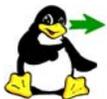
SCSI란 콤퓨터에 주변장치를 접속하기 위한 직렬표준이음부로서 입출력모선을 접속하는데 필요한 기계적 및 전기적 요구사항과 주변장치를 중심으로 한 지령묶음에 대한 규격을 말한다. SCSI는 주변장치의 번호만 각각 지정해주면 자료의 충돌문제가 제기되지 않기때문에 널리 사용되고있다. 그러나 가격이 비싸고 주변장치자체가 사용하는 규약과 조금이라도 다르면 사용할수 없는 부족점이 있다.

env

환경변수보기

지령형식: env [추가선택] [변수=값...] [지령[인수...]]

-, -i, --ignore-environment : 현재환경을 무시하고 지정한 "변수=값"으로 한다.
-u, --unset=name : 지정한 변수를 제거한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

env지령은 현재 설정된 환경변수를 보여주기도 하고 새로운 값으로 설정한 다음 변경된 환경을 보여주기도 한다.

련관지령

export 환경변수를 확인/수정하는 지령

ex

행편집기

ex는 행단위로 편집하는 대화식본문편집기로서 지금은 그리 많이 쓰이지 않으나 vi편집기에 내장되여있으므로 ex편집기의 여러 기능이 vi를 통하여 실현된다. 그러므로 ex편집기의 기능은 vi에서 설명한다.

ed에서 vi까지

ed	UNIX의 행단위편집기
ex	발전된 형태의 ed(Extended ed)
vi	화면편집기
vi Clone	vi의 발전된 형태로서 대표적인것은 vim, elvis를 들수 있다.

핵심부에 자기 이름 넣기

uname -a 지령으로 Linux 배포판들의 핵심부판본정보를 확인해보면 2.4.20-8 혹은 2.4.20-1hl과 같이 핵심부의 판본뒤에 추가사항이 붙은것을 볼수 있다.

핵심부를 콤파일할 때 핵심부정보에 자기의 성이나 임의의 단어 등을 넣을수 있다.

핵심부원천에서 Makefile이라는 파일을 열어보면 "EXTRAVERSION=" 부분이 있다. 이 부분뒤에 추가하는 내용이 핵심부정보에 덧붙는다.

다음과 같이 입력하고 파일을 보관한다.

```
EXTRAVERSION=hana
```

핵심부를 콤파일한 후에 uname -a지령으로 핵심부판본정보를 확인해보면 2.4.20-hana로 표시된다.

expand

/usr/bin/expand

문서변환

지령형식: expand [추가선택] [파일이름 ...]

-t, --tabs=NUMBER : 태브의 간격을 조절한다. 기정으로는 8칸으로 되여있으나 지정한 수자에 따라 태브의 간격을 조정할수 있다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일안에 있는 태브를 적당한 수의 공백으로 변환해주는 지령이다.

실례로 아래와 같은 내용을 가진 test.txt파일이 있다고 하자. 3번째 행의 1234와 56사이 그리고 4번째 행의 123, 456, 789사이에 태브가 있다.

이제 expand지령으로 태브의 간격을 조절해보자. 태브의 간격을 한번은 3으로, 다음에는 10으로 지정한다.

하위등록부안에 있는 파일내용을 모두 수정하기

여러개 파일안에 있는 "계산기"라는 글자를 "콤퓨터"라고 수정할 때 하위등록부에 있는 파일들까지 모두 수정할수 있다면 매우 편리할것이다.

그러면 먼저 "계산기"라는 문자가 들어있는가를 하위등록부의 파일들까지 모두 검색해보자.

```
grep 계산기 * -rls
```

일정한 문자렬을 다른 문자렬로 변경할 때에는 vi를 사용하여 다음과 같이 할수 있다. -c추가선택을 사용하면 파일을 열면서 동시에 지시한 vi지령을 수행한다.

```
vi -c :%s/계산기/콤퓨터/g
```

이 두가지 지령을 한번에 련결하여 수행해보자. 다음과 같이 for문을 리용할수 있다.

```
for i in 'grep 계산기 * -rls' ; do (vi -c :%s/계산기/콤퓨터/g -c :wq $i); done
```

expr

/usr/bin/expr

표현식평가

지정한 인수를 표현식으로 인식하는 쉘스크립트를 만들 때 참고하는 지령이다.

자세한 사용법은 쉘스크립트를 참고하기 바란다.

실레로 이 지령은 주어진 인수를 계산가능한 표현식으로 인식하여 계산한다. 간단히 쉘상에서 더하기계산을 해보자.

주의 표현식을 이루는 항목 하나하나를 띄여써야 한다. 즉 "1 + 2"가 아니라 "1+2"로 쓰면 문자렬로 인식해버린다.

false

/bin/false

항상실패

지령형식: false

false [--help, --version]

--help : 도움말을 출력한다.
--version : 판본정보를 출력한다.

주로 쉘스크립트에서 사용하며 아무것도 하지 않고 항상 실패 혹은 거짓을 되돌린다. 자세한 사용법은 쉘스크립트에서 설명한다.

xhost 활용하기

두대이상의 콤퓨터를 사용하는 경우에 한 콤퓨터에서 다른 콤퓨터의 X프로그람들을 사용할수 있다.

이 콤퓨터들의 IP를 각각 192.168.0.2과 192.168.0.3이라고 가정하자. 먼저 192.168.0.3인 콤퓨터에서 192.168.0.2인 콤퓨터로 원격접속한다.

```
# ssh root@192.168.0.2
```

접속이 끝나면 DISPLAY환경변수를 192.168.0.3:0.0으로 설정하여 다음에 실행되는 프로그람들을 192.168.0.3인 콤퓨터에서 보게 한다.

```
# ssh root@192.168.0.2
```

fdformat

/usr/bin/fdformat

플로피디스크 형식화

지령형식: fdformat [-n] 장치이름

-n : 형식화를 진행한 다음 디스크표면검사를 하지 않는다.

Linux에서 플로피디스크를 형식화하는 지령이다.

Linux에서 플로피디스크를 가리키는 장치이름은 /dev/fd0, /dev/fd1이다. 형식화할 때에는 용량을 지정해야 하는데 만일 1.44MB의 플로피디스크를 형식화하려면 장치이름을 /dev/fd0H1440으로 지정해야 한다.

체계가 인식한 장치이름을 보려면 dmesg지령으로 쉽게 알아볼수 있다.

현재 체계가 인식하고있는 플로피디스크장치이름이 fd0이므로 그것을 리용하여 플로피디스크를 형식화해보자.

플로피디스크를 넣고 /dev/fd0을 형식화한다.

플로피디스크장치가 1.44MB용이므로 장치이름을 /dev/fd0H1440이라고 지정해도 된다.

ls /dev/fd0*지령으로 플로피디스크에 해당한 장치들을 볼수 있는데 이중에서 제일 많이 사용하는것은 /dev/fd0H1440이다.

fdisk

/sbin/fdisk

구획설정

지령형식: fdisk [추가선택] [장치이름]

-v: 판본정보를 보여준다.
-l: 현재체계의 구획표를 보여준다.
-s 구획 : DOS구획이 아닌 구획의 크기를 보여준다.

구획을 설정하는 지령이다. Linux를 설치할 때 구획설정은 보통 Disk Druid와 fdisk, cfdisk로 한다. Disk Druid를 리용하는 경우에는 대부분 간단히 설정할수 있지만 구획설정을 세부적으로 해야 할 때에는 fdisk를 리용한다.

그러면 Linux상에서 /dev/hda장치를 fdisk지령으로 보도록 하자. -l추가선택을 리용하면 지정한 구획의 정보를 볼수 있다.

```
[root@Dolphin root]# fdisk -l /dev/hda

Disk /dev/hda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1       382   3061768+   b  Win95 FAT32
Partition 1 does not end on cylinder boundary.
/dev/hda2          1021      1532   4105080    c  Win95 FAT32 (LBA)
/dev/hda3          1532      4865  26777520    f  Win95 Ext'd (LBA)
/dev/hda4           382      1021   5133240   83  Linux
Partition 4 does not end on cylinder boundary.
/dev/hda5          1532      4810  26331448+   b  Win95 FAT32
/dev/hda6          4810      4865    446008+  82  Linux swap

Partition table entries are not in disk order
```

-s추가선택을 리용하면 지정한 매 구획의 크기를 알아볼수 있다.

```
[root@Dolphin root]# fdisk -s /dev/hda
39082680
[root@Dolphin root]# fdisk -s /dev/hda1
3061768
[root@Dolphin root]# fdisk -s /dev/hda2
4105080
```

우에서는 현재 하드디스크에 설정된 구획을 보기만 하였다. 이제는 fdisk를 리용하여 /dev/hda에 대한 구획을 설정하는 방법을 보도록 하자.

```
[root@Dolphin root]# fdisk /dev/hda

The number of cylinders for this disk is set to 4865.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

이제 m을 입력하면 fdisk지령에 대한 사용법을 볼수 있다.

p라는 지령을 입력하면 현재하드디스크의 구획표를 볼수 있다.

command(m for help): 행에 입력할수 있는 지령들은 다음과 같다.

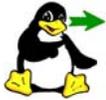
- a 기동할수 있는 기발로 고정
- b bsd디스크이름을 편집
- c DOS호환기발로 변경
- d 구획을 삭제
- l 이미 알려져있는 구획류형들의 목록
- m 지령들에 대한 설명을 출력
- n 새로운 구획을 추가
- o 빈 DOS구획표를 새로 작성
- p 구획표를 출력
- q 변경된 내용을 저장하지 않고 완료
- s 빈 Sun디스크이름을 새로 생성
- t 구획의 체계id를 변경
- u 표시/입구 단위를 변경
- v 구획표를 검사
- w 디스크에 구획표를 기록하고 완료
- x 기타 기능

그러면 fdisk사용법을 실례를 들어 자세히 보기로 한다.
20GB 하드디스크를 아래의 2가지 경우로 나누어보자.

Windows 6GB, Linux 6GB, 나머지 령역은 Windows여벌복사령역으로 설정할 경우

① Windows상에서 fdisk지령으로 약 6GB정도의 구획을 설정한 다음에 설치한다. (/dev/hda1에 6GB용량의 구획설정)
② Linux설치도중에 구획설정에서 fdisk로 들어간다.
③ "p"를 입력하여 구획정보를 본다.
④ hda1을 제외한 모든 령역을 삭제한다.
⑤ 새로운 구획을 생성한 다음 구획류형을 지정한다.
n → p → 2 → → +6000M
⑥ /dev/hda3령역에 교환령역으로 128MB를 생성한다.
n → p → 3 → → +128M → t → 82
⑦ /dev/hda4 령역에 나머지 남은 디스크용량을 모두 여벌복사령역으로 만든다.
n → p → 4 → → → t → c
⑧ 구획설정을 보관(w)한 다음 Linux를 설치한다.

20GB 하드디스크에 모두 Linux를 설치할 경우

① Linux 설치도중에 구획설정에서 fdisk로 들어간다.
② "p"를 입력하여 구획정보를 본 후 모두 삭제한다.
③ /dev/hdal 령역에 교환령역으로 128MB를 생성한다.
   n → p → 1 → → +128M → t → 1 → 82
④ 나머지 용량모두를 Linux령역으로 생성한다.(/dev/hda2령역)
   n → p → 2 → →
⑤ 구획설정을 보관(w)한 다음 Linux를 설치한다.

련관지령

cfdisk DOS형식의 구획설정지령

fetchmail

usr/bin/fetchmail

우편물꺼내기

지령형식: fetchmail [추가선택] [우편봉사기]

-V, --version : 판본정보를 보여준다.
-c, --check : 우편물을 가져오거나 삭제하지 않고 대기하는 상태
-s, --silent : 통보문을 표시하지 않는다.
-v, --verbose : 진행상태를 상세히 보여준다.
-u, --username : 우편봉사기에 접속할 사용자계정을 지정한다.
-p, --protocol : 규약을 지정한다. 지정할수 있는 규약은 다음과 같다. AUTO, POP2, POP3, APOP, RPOP, KPOP, SDPS, IMAP, ETRN, ODMR
-K, --nokeep : 우편물을 검색한 후 새로운 통보문을 봉사기에서 삭제한다.
-k, --keep : 우편물을 검색한 후 새로운 통보문을 봉사기에 저장한다.

fetchmail은 봉사기에서 우편물을 검색하여 가져오는 편의프로그람이다.

POP2, POP3, IMAP2bis, IMAP4, IMAPrev1과 같은 규약을 통해 봉사기에서 우편물을 가져오고 국부콤퓨터에 있는 sendmail지령을 통하여 통보문을 25포구로 전달한다.

먼저 mail지령으로 우편물을 보낸다.

보낸 우편물을 국부콤퓨터에 보관하려면 다음과 같이 한다.

```
# fetchmail -p pop3 -u admin -k mail.linuxroot.sec.com
```

그러면 규약을 pop3으로 지정하며 검색된 우편은 POP3봉사기에 보관해둔다. 이때 새로운 통보문은 /var/spool/mail/root파일로 보관된다.

우편봉사기에 있는 우편물을 다른 계정으로 옮기기

용어 상식

Linux봉사기에 도착한 우편물을 다른 우편계정으로 옮기고 싶을 때 우편물을 손쉽게 옮겨주는 방법이다.

만일 linuxroot.sec.com봉사기의 han계정에 온 우편물을 admin계정으로 옮기려면 다음과 같은 지령을 실행한다. 우편봉사기에 도착하는 우편물은 /var/spool/mail등록부의 해당한 계정등록부에 저장된다.

```
# cat /var/spool/mail/han | formail -s /usr/sbin/sendmail admin@linuxroot.sec.com
```

이와 같은 작업을 자동적으로 하려면 다음과 같이 .forward라는 숨김파일을 만든다. 자기에게 오는 모든 우편물을 admin@linuxroot.sec.com이라는 계정으로 보내려면 .forward라는 이름의 파일을 만들고 이 우편주소만 쓰면 된다.

동시에 여러 계정으로 보낼 때에는 다음과 같이 입력하고 저장한다.

```
admin@linuxroot.sec.com
happy@happylinuxing.com
pretty@linuxroot.co.kp
```

file

/usr/bin/file

파일종류 확인

지령형식: file [추가선택]… 파일…

-b : 지정한 파일의 이름은 출력하지 않고 류형만을 보여준다.
-f filelist : 파일목록에서 지정한 파일들에 대해 지령을 실행한다.
-i : 일반형식이 아니라 MIME형식의 문자렬로 보여준다.(실례로 ASCII본문파일이라면 text/plain; charset-us-ascii형태로 보여준다.)
-m magic파일 : 기정으로 지정된 /etc/magic를 사용하지 않고 magic파일을 지정한다.
-L : 기호련결파일들만 검색한다.
-z, --uncompress : 압축된 파일의 내용을 보여준다.
--help : 도움말정보를 보여준다.
--version : 판본정보를 보여준다.

file지령은 파일의 종류를 구분하는 편의프로그람으로서 file지령뒤에 파일이름을 입력하면 ASCII나 text, date와 같은 파일의 종류를 알려준다. /usr/share/magic 혹은 /usr/lib/magic라는 파일에 여러가지 파일종류에 대한 정보가 포함되여있으므로 그에 기초하여 파일의 종류를 알수 있다.

아래에 보여준 실례는 test파일의 파일류형을 알려준다.

-b추가선택을 사용하면 파일이름은 보여주지 않고 파일류형만 볼수 있다.

-i추가선택은 파일의 류형을 MIME형식으로 볼수 있다.

MIME

ASCII자료만 처리할수 있는 원래의 Internet전자우편규약이다. 즉 SMTP를 확장하여 음향, 동화상, 화상, 응용프로그람 기타 여러가지 종류의 자료파일들을 주고받을수 있도록 기능이 확장된 규약이다.

find

/usr/bin/find

파일찾기

지령형식: find [경로] [추가선택] [작업]

-name name : 지정된 이름의 파일을 찾는다.
-user name : 지정된 사용자가 소유하고있는 파일을 찾는다.
-type [bcdfls] : 지정된 형식의 파일을 찾는다.

- b 블로크파일
- c 문자
- d 등록부
- f 파일
- l 련결파일
- s 소케트

-size [+/-]n [bckw] : 지정된 크기의 파일을 찾는다.

- +n n보다 크다.
- -n n보다 작다.
- n n이다.
- b 512byte
- c byte
- k kilobytes
- w 2byte

-inum number : 지정한 inode번호를 가진 파일을 찾는다.
-print : 검색된 파일이름을 표준출력에 현시한다.
-exec command{}\; : 찾은 매 파일에 대해 지정된 지령을 실행한다.
-ok command{}\; : 실행여부를 사용자에게 확인한 후 지령을 실행한다.

find지령은 Linux에서 주어진 조건에 따라 등록부의 구조를 검색하여 해당한 파일을

찾는 지령이다. 흔히 find는 파일검색보다는 선택조건에 일치하는 파일을 찾아서 지령을 실행하는데 많이 리용한다.

먼저 이름을 지적하여 파일을 찾아보자.

find는 검색할 경로를 추가선택앞에 미리 정의하여야 한다. 실례로 root등록부에서 word.awk라는 파일을 검색해보자. 이 경우에는 -name추가선택을 사용해야 한다.

exec command \;은 find가 찾아낸 매 파일에 대하여 command를 실행한다. 이때 command는 "\;"으로 끝나야 한다.

현재 등록부에 있는 모든 파일이름을 현시

```
# find -print
```

현재 등록부에 있는 하위등록부를 현시

```
# find -type d -print
```

현재 등록부에서 파일이름에 공백문자가 들어간 파일들을 모두 삭제

```
# find -name "* *" -exec rm -f{}\;
```

현재등록부에 있는 파일들중에서 파일이름에 비정상적인 글자가 있는 파일을 삭제

주의 +{;"\=?~()<>&*|$ 등의 특수기호를 포함한 파일을 삭제한다. 이 과정에 2>/dev/null의 오유출력은 무시된다.

요구하는 파일체계를 모두 복사

```
# mkdir /hda1
# mount /dev/hda1 /hda1
# cd /home
# find -depth -print | cpio -pmdvl /hda1
```

지난 24시간동안에 변경된 모든 파일목록을 changelist파일로 작성

```
# find / -mtime -1 \! -type d -print > changelist
```

finger

/usr/bin/finger

사용자정보 보기

지령형식: finger [-lmsp] [user...] [user@host...]

-l : 사용자의 home등록부, 집전화번호, 등록가입쉘, 우편상태, home등록부파일 등과 함께 -s추가선택으로 보이는 정보를 여러 행 형식으로 보여준다.

-s : 사용자의 등록가입이름, 실제이름, 말단이름, 상태, 유휴시간, 등록가입시간, 사무실위치, 사무실전화번호를 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# cat /etc/xinetd.d/finger
# default: on
# description: The finger server answers finger requests. Finger is \
#       a protocol that allows remote users to see information such \
#       as login name and last login time for local users.
service finger
{
        socket_type     = stream
        wait            = no
        user            = nobody
        server          = /usr/sbin/in.fingerd
        disable         = yes
}
```

finger는 지정한 사용자정보를 알려주는 지령이다. 편리한 기능이기는 하지만 침입자에게 이 정보가 악용될수 있으므로 될수록 finger설정은 하지 않는것이 좋다. Red Hat 7.2부터는 finger설정을 /etc/xinetd.d/finger파일에서 한다.

/etc/xinetd.d/finger파일에서 disable=no로 설정한 다음 xinetd데몬을 재실행한다.

free

/usr/bin/free

기억기정보 보기

지령형식: free [-b | -k | -m][-o][-s delay][-t][-V]

-b : 기억기의 용량을 byte로 표시한다.
-k : KB단위로 표시한다.
-m : MB단위로 표시한다.
-t : 총계가 포함된 행을 출력한다.
-o : 완충기에 대한 정보는 현시하지 않는다.
-s delay : 지정된 시간(s)마다 련속 출력한다. 시간간격값으로 실수값도 줄수 있다.

free지령은 기억기의 사용정형을 알아보는 지령으로서 기억기와 교환령역, 핵심부가 사용하는 공유기억기와 완충기에 대한 사용상태를 모두 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# free
             total       used       free     shared    buffers     cached
Mem:        247184     178464      68720          0       9708      49248
-/+ buffers/cache:     119508     127676
Swap:       446000      86856     359144
```

free -s추가선택으로는 s단위로 변하는 기억기의 사용상태를 알수 있다.

```
root@Dolphin:~
[root@Dolphin root]# free -s 1
             total       used       free     shared    buffers     cached
Mem:        247184     182272      64912          0       9748      51200
-/+ buffers/cache:     121324     125860
Swap:       446000      86432     359568

             total       used       free     shared    buffers     cached
Mem:        247184     182272      64912          0       9748      51200
-/+ buffers/cache:     121324     125860
Swap:       446000      86432     359568

             total       used       free     shared    buffers     cached
Mem:        247184     182272      64912          0       9748      51200
-/+ buffers/cache:     121324     125860
Swap:       446000      86432     359568
```

합리적인 방법으로 교환구획 설정

사용자들은 보통 교환구획을 기억기의 2~3배정도의 크기로 만든다. 그러나 보다 과학적이고 합리적인 방법을 아는것이 필요하다. 교환령역은 모자라서는 안되지만 너무 크게 만드는것은 랑비이다.

먼저 필요한 최대기억용량을 확인한다. 즉 한번에 구동하려 하는 모든 프로그람이 요구하는 기억기의 총량을 알아본다. 실례로 XWindows환경은 대략 8MB이상의 기억기가 필요하며 KDE환경에서는 좀 더 많이 필요하므로 교환령역을 넉넉하게 해주어야 한다. gcc로 프로그람을 콤파일할 때는 보통 4MB정도가 필요하지만 프로그람의 크기에 따라 수십MB이상의 기억기가 필요할수도 있다.

핵심부는 그자체가 1MB정도를 차지하며 일반적인 쉘과 작은 편의프로그람 역시 다 합쳐서 1~2MB정도 차지한다. 또한 같은 프로그람을 동시에 여러개 실행하거나 혹은 서로 다른 사용자가 각각 사용하고있을 경우에도 실행코드와 공유서고의 용량을 제외한 나머지부분은 각각 계산해주어야 한다. 이렇게 계산한 용량의 10~20%정도를 여유용량으로 더해준다. 여기에서 실제 기억기의 용량을 빼고 남은 값이 교환령역의 합리적인 크기로 된다. 만일 이렇게 얻어낸 교환령역이 기억기의 2~3배가 넘는다면 기억기를 더 확보해야 한다.

fsck

/sbin/fsck

파일체계점검

지령형식: fsck [-AVRTNP][-s] [-t 파일체계류형] [추가선택] 파일체계 [...]

-f : 파일체계에 이상이 없어도 무조건 검사하게 한다.
-A : /etc/fstab에 등록된 내용을 참고하여 체계에서 사용하는 모든 파일체계에 대한 검사를 진행한다.
-a : fsck가 검사하는 도중에 오유가 발생하면 이것을 자동으로 처리하도록 하는 추가선택이다. 검사도중에 현시되는 질문에 일일이 대답하기가 귀찮거나 모두 y로 처리해도 무방한 경우에 사용한다.
-b : 슈퍼블로크가 손상되였을 경우 사용하는 추가선택이다. 슈퍼블로크는 첫번째 블로크에

위치하는데 만일 슈퍼블로크가 파괴되였다면 매 8192블로크마다 여벌복사된 슈퍼블로크의 복사본을 가지고 슈퍼블로크를 복구해야 한다.

-R : -A추가선택과 같이 사용될 때 root파일체계는 검사에서 제외한다.(이미 읽기-쓰기가능으로 탑재되여있는 경우)

-T : 시작할 때 제목을 보여주지 않는다.

-N : 실행하지는 않고 어떤 작업을 하겠는가만 보여준다.

-P : -A추가선택이 설정되여있을 때 다른 파일체계와 root파일체계를 병렬처리하여 점검한다. root파일체계에 오유가 있는 경우에는 e2fscsk지령의 실행파일과 같은것들이 파괴되여있을수 있기때문에 안전한 방법이 못된다. root파일체계의 구획을 좀 더 작게 수정할 필요가 없는 체계관리자들을 위해서만 제공되는 추가선택이다.

-S : fsck지령을 직렬화한다. 대화형방식에서 여러 파일체계를 점검할 때 리용한다.(주의 : e2fsck는 기정으로서 대화형으로 실행된다. e2fsck를 비대화형방식으로 실행하려면 -P 또는 -a추가선택을 주어 오유를 자동교정하거나 -n추가선택을 주어 오유를 고치지 않고 그냥 넘어가도록 해야 한다.)

-v : 실행되는 매 지령에 대한 구체적인 설명을 준다.

-t type : 점검할 파일체계류형을 지정한다. -A추가선택이 설정되여있는 경우 파일체계류형에 맞는것만 점검한다. 파일체계류형앞에 no를 붙이면 그 파일체계류형을 제외한 나머지 류형들에 대하여 점검한다. 보통 /etc/fstab파일에 들어있는 파일체계류형에 따른다. 만약 해당 류형을 알수 없을 때에는 -t추가선택에서 주어진 값을 사용한다. 주어진 값이 유효하지 않다면 기정값인 ext2가 사용된다.

파일체계를 점검하고 문제가 생긴 부분은 수복하는 지령으로서 기동할 때 자동으로 파일체계를 점검한다.

일반적으로 Linux체계가 기동할 때 자동적으로 파일체계를 검사하고 자동복구하는것은 /etc/rc.d/rc.sysinit파일에서 fsck를 설정해놓았기때문이다. 그러므로 수동으로 검사할 필요는 거의나 없다. 만일 검사하는 과정에 inode가 손상된 파일이 발견되면 그 파일을 /lost+found등록부에 옮겨놓는다.

/dev/hda1파일체계에 대한 검사를 진행하려면 다음과 같은 지령을 리용한다.

```
# fsck -f -v /dev/hda1
```

련관지령

ext2　　파일체계점검. tsct -t ext 지령과 같다.

새로운 하드디스크 탑재하기

용어 상식

현재 사용하고있는 하드디스크는 Primary Master(/dev/hda)에 탑재되여있으며 추가할 하드디스크는 Primary Slave(/dev/hdb)에 탑재되였다고 하자.

fdisk 실행

다음과 같은 방법으로 새 하드디스크에 Linux구획을 설정한다.

```
# fdisk /dev/hdb
```

구획정보 확인

구획정보를 확인한다. 새로운 하드디스크를 탑재할 때에는 아래와 같이 구획정보가 나타나지 않는다.

```
Command (m for help): p
Disk /dev/hda:255 heads, 63 sectors, 788 cylinders
Units = cylinders of 16065 * 512 bytes
Units = cylinders of 16065 * 512 bytes
Device Boot Strat End Blocks Id System
```

새 구획의 생성

새로운 구획을 생성한다. 아래의 내용은 구획을 하나로 설정하는 경우이다.

```
Command (m for help): n
Command action
e extended p primary patition (1-4)
p
(구획을 Primary로 설정한다.)
Patition number (1-4): 1
(사용할 구획을 수값으로 입력한다.)
First cylinder (1-788, default 1):
Using default value 1
(초기값을 입력한다.)
Last cylinder or +sizeM or +sizeK(1-788, default 788):
Using default value 788
(마지막 실린더값을 입력한다.)
```

새 구획정보 확인

```
Command (m for help): p
Disk /dev/hda:255 heads, 63 sectors, 788 cylinders
Units = cylinders of 16065 * 512 bytes
device Boot Start End Blocks Id System
/dev/hdb 1 1 788 6329578+ 83 Linux
```

보관하고 완료

구획정보를 보관한 다음 완료한다.

```
Command (m for help): w
```

파일체계 생성

체계를 재기동한 후 분할된 구획을 ext2 파일체계로 생성한다.

```
# mke2fs /dev/hdb1 또는 # mkfs -t ext2 /dev/hdb1
```

탑재

파일체계를 생성한 다음 /hdbl에 탑재한다.

```
# mkdir /hdbl
# mount /dev/hdb1 /hdb1
```

ftp

/usr/bin/ftp

ftp봉사기접속 의뢰기프로그람

지령형식: ftp [-v][-d][-i][-n][-g][-k realm][-f][-x][-u][-t][콤퓨터이름]

-v : 접속과정을 자세히 보여준다.

ftp봉사기에 접속하기 위한 ftp의뢰기이다.
접속 후 ftp>다음에 사용할수 있는 지령들은 다음과 같다.

- ? : 도움말정보를 보여준다.
- ascii mode : ASCII방식과 같다
- binary mode : 2진방식과 같다.
- connect 콤퓨터 [포구] : 지정한 콤퓨터와 포구로 접속을 시도한다.
- get 파일이름 : 원격접속콤퓨터의 파일을 국부체계에 내리적재한다.
- get 파일이름 국부파일이름 : 원격접속콤퓨터의 파일을 지정한 국부파일이름으로 내리적재한다.
- get 파일이름1 파일이름2 파일이름3...파일이름N : 지정한 파일들을 내리적재한다.
- mode 전송방식 : ASCII방식이나 2진방식으로 설정한다.
- put 파일이름 : 국부콤퓨터에 있는 파일을 원격콤퓨터에 올리적재한다.
- put 국부파일이름 원격파일이름 : 국부파일이름을 원격콤퓨터에 지정한 원격파일이름으로 올리적재한다.
- put 파일이름1 파일이름2 파일이름3 ... 파일이름N : 지정한 파일들을 원격콤퓨터의 등록부에 올리적재한다.
- quit : ftp접속을 끝낸다.

ftp봉사기접속방법

linuxroot.co.kp콤퓨터에 접근하는 방법에는 ftp지령뒤에 콤퓨터이름을 함께 입력하는 방법과 ftp를 실행하고 open지령으로 콤퓨터에 접속하는 방법이 있다.

```
# ftp linuxroot.co.kp
```

또는

```
# ftp
ftp> open linuxroot.co.kp
```

ftp봉사기에 접속하기

접속을 시도하면 ftp봉사기프로그람의 상태정보가 나타나며 접속하려고 하는 사용자계정을 입력하라는 재촉문이 나온다. 만일 계정을 입력하지 않으면 anonymous계정으로 접속하게 된다.

admin계정을 입력하고 계속하여 통과어를 입력하여 접속을 진행한다.

파일전송류형

ftp파일전송의 기본방식은 ascii로써 ASCII파일전송에 사용된다. 2진방식으로 전환하려면 bin지령을 리용한다.

ftp봉사기목록 보기

ls지령으로 ftp봉사기의 목록을 본다.

```
ftp> ls
227 Entering Passive Mode (211,255,253,59,7,240).
150 Opening ASCII mode data connection for file list
drwxr-xr-x  3  admin  admin  4096     Dec 14 13:32  Desktop
drwxr-xr-x  2  admin  admin  4096     Dec 14 13:01  Desktop.back
-rw-r--r--  1  admin  admin  0        Feb  2 10:16  Recycled
-rw-r--r--  1  admin  admin  6688768 Jul  4 2001   SSHWinClient-3.0.0.exe
-wr-r--r--  1  admin  admin  0        Feb  2 10:16  _Restore
drwxr-xr-x  4  admin  admin  4096     Dec 14 13:01  kde.back
drwxrwxr-x  2  admin  admin  4.96     Mar 26 11:33  movies
```

등록부 이동하기

cd지령을 리용하면 등록부를 이동할수 있다. 다음의 화면에서 "250"으로 시작하는 통보문은 봉사기가 보내는 정보통보문이다.

파일 받기와 보내기

파일의 받기는 "get 봉사기파일이름", 보내기는 "put 국부파일이름"이라는 형식으로 지령을 입력하면 된다. 받거나 보내는 파일의 개수가 많으면 mget, mput지령을 쓴다.

이 상태에서 국부등록부를 변경하려면 !cd지령을 리용한다.

련관지령

ncftp ftp봉사기접속 의뢰기프로그람.

Hash mark

ftp로 파일을 내리적재하는동안 그 과정을 보고싶다면 hash지령을 쓴다. ftp는 내리적재받는 자료가 1024byte로 될 때마다 Hash mark를 보여준다.

ftpcount

/usr/bin/ftpcount

ftp를 사용하는 인원 확인

지령형식: ftpcount [-V]

-V : 저작권과 판본정보를 보여준다.

ftpcount지령은 ftpaccess파일에서 정의된 매 class에 대한 사용자수 또는 제한된 사용자의 접속회수와 ftp봉사기의 PID를 보여준다.

대기렬(queue)

자료가 들어간 순서대로 처리되는 자료구조를 말한다. 이러한 순서를 가리켜 흔히 선입선출(FIFO: first in first out)이라고 한다. 이와는 반대로 탄창자료구조는 먼저 들어간 자료가 제일 마지막에 처리되는 형식으로서 후입선출(LIFO: last in first out)이라고 한다.

ftpwho
/usr/bin/ftpwho

ftp접속자보기

 지령형식: ftpwho [-V]

-V : 저작권과 판본정보를 보여준다.

ftp에 접속해있는 사람과 접속주소를 확인한다.

ftp봉사기를 운영하거나 Linux에서 개별적으로 ftp봉사기를 가동한 경우 현재 시점에서 체계에 접근한 ftp사용자를 확인해야 할 경우가 있다. 이 경우에 ftpwho지령을 사용하면 간단하게 확인할수 있다.

아래에서는 매 ftp사용자에 대한 프로세스정보를 보여준다. 현재 admin계정이 127.0.0.1의 IP로 접속했다는것을 알수 있다.

fuser
/usr/bin/fuser

프로세스식별

 지령형식: fuser [추가선택]

보안을 위해 프로세스를 식별하고 통제하는 지령이다.

어떤 파일이나, 파일집합, 등록부에 접근하고있는 프로세스를 PID로 식별해준다. -k추가선택을 쓰면 해당 프로세스를 완료할수 있다. 이 지령은 주로 스크립트에서 체계봉사에 대하여 허가받지 않은 사용자가 접근하는것을 막는데 리용된다.

실례로 탑재된 CD-ROM에 대하여 umount지령을 사용하면 "Resource busy"와 같은 오유통보문이 나타나는것을 볼수 있다. 이때 -k추가선택은 /mnt/cdrom등록부에 있는 해당 프로세스를 강제로 완료시킨다.

TX

Transmit Data의 략자로서 망에서 자료전송을 의미한다.

RX

Receive Data의 략자로서 망에서 자료수신을 의미한다.

grep

/bin/grep

문자렬검색

 지령형식: grep [추가선택] 패턴 [파일 ...]

-A num, --after-context=num : 일치하는 행부터 그뒤로 지정한 행수만큼의 내용들을 보여준다.
-b, --byte-offset : 매 행을 보여주기전에 입력파일의 편위바이트를 보여준다.
-B num, --befole-context=num : 일치하는 행에서 앞으로 지정한 행수만큼의 내용을 보여준다.
-c, --count : 일치하는 행의 개수를 보여준다.
-C[num], --before-context=num : 일치하는 행의 우와 아래의 내용을 num행만큼 보여준다.
기정값은 두행이다.
-d action, --directories=action : 등록부인 경우 지정한 동작을 실행한다. 기정 동작은 read이다.
아래에 실행할수 있는 동작들을 보여준다.

- read : 등록부를 보통의 파일처럼 읽는다.
- skip : 등록부를 건너뛴다.
- recurse : 등록부안에 있는 모든 파일들을 읽는다.

-e pattern, --regexp=pattern : 하나이상의 탐색패턴을 지정하며 단순한 패턴과 같이 동작한다.
-f file, --file=FILE : 지정한 파일로부터 패턴을 얻는다.
-h, --no-filename : 결과목록에서 파일이름은 보여주지 않는다.

-i, --ignore-case : 대소문자를 구별하지 않는다.
-l, --files-with-matches : 일치하는 행이 있는 파일이름만 보여주고 그 행의 내용은 보여주지 않는다.
-n, --line-number : 일치하는 행과 파일에서 그 행이 몇번째 행인가를 보여준다.
-q, --quiet, --silent : 결과를 보여주지 않는다.
-r, --recursive : 매 등록부안에 있는 파일들까지도 읽는다. -d recurse추가선택과 같다.
-s, —no-message : 존재하지 않거나 읽을수 없는 파일에 대하여 오유통보문을 보여주지 않는다.
-v, --revert-match : 지정한 패턴과 일치하지 않는것들을 보여준다.
-w, --word-regexp : 한 단어로 일치하는 결과만을 보여준다.
-x, --line-regexp : 패턴과 일치하는 전체 행수를 보여준다.
-y : -i추가선택과 같다.

입력되는 내용에서 지정한 문자렬을 검색하여 그 문자렬이 들어있는 행을 화면에 출력하는 지령이다.

련관지령

egrep = grep -E
fgrep = grep -F

현재등록부에 있는 하위등록부만 보기

```
# ls -la | grep "^d"        # 숨은 등록부까지 보기
# ls -l | grep "^d"         # 일반등록부만 보기
```

alias지령으로 편리하게 사용할수도 있다.

```
# alias la='ls -la | grep "^d"'
```

압축된 파일에서 검색

zerep, zegrep, zigrep는 압축되지 않은 파일에 대하여 문자렬검색을 진행하지만 grep, egrep, fgrep보다는 약간 느리다. bzip로 압축된 파일에서 검색하려면 bzgrep를 리용한다.

groupadd

/usr/sbin/groupadd

그룹추가

지령형식: groupadd [추가선택] 그룹

-g gid : 생성하는 그룹의 그룹ID를 지정한다.
-o : -g추가선택과 함께 사용하며 지정한 그룹ID가 이미 있더라도 중복을 허용한다.
-r : 체계관리령역인 499이하의 그룹ID에서 그룹을 추가한다.

groupadd지령은 새로운 그룹을 추가하는 지령으로서 /etc/group파일과 관련이 있다.

adm그룹에는 root, adm, daemon사용자가 속해있다는것을 알수 있다.

련관지령

groupdel　그룹을 삭제한다.
groupmod　그룹의 정보를 변경한다.
groups　매 사용자가 속한 그룹을 보여준다.

groupdel

/usr/sbin/groupdel

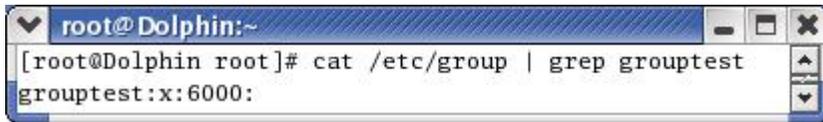
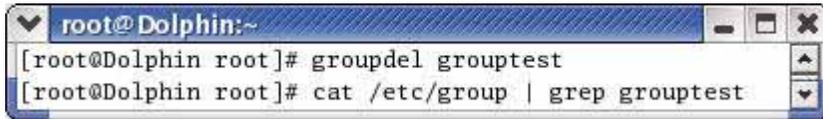
그룹삭제

지령형식: groupdel 그룹이름

groupadd지령으로 추가된 그룹을 삭제할수 있다. group지령 다음에 삭제하려고 하는 그룹이름을 지정하면 된다.

먼저 groupadd지령으로 grouptest그룹을 추가해보자.

```
# groupadd grouptest
```

/etc/group파일에서 grouptest가 있는 행을 grep지령으로 추출한 결과 아래와 같이 grouptest그룹이 추가되였다는것을 알수 있다.

추가한 grouptest그룹을 groupdel지령으로 삭제한다.

groupmod

/usr/bin/groupmod

그룹정보수정

지령형식: groupmod [추가선택] 그룹

-g gid : 그룹의 ID를 지정한 gid로 변경한다.
-n name : 지정한 name으로 그룹이름을 변경한다.
-o : -g추가선택과 함께 사용하며 지정한 gid가 이미 존재하여도 중복을 허용한다.

groupadd지령으로 추가된 그룹의 이름을 변경할 때 리용하는 지령이다.

현재 /etc/group파일에 groupadd지령에서 추가한 grouptest그룹이 추가되여있다고 하자.

-n추가선택을 리용하여 grouptest그룹이름을 grouptest1로 변경할수 있다.

이외에도 -g추가선택을 지정하여 현재 6000으로 되여있는 gid를 변경할수 있다.

```
# groupmod -g 6001 grouptest1
```

groups

/usr/bin/groups

사용자그룹보기

 지령형식: groups [추가선택] [사용자]

--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

사용자가 속한 그룹의 이름을 보여주는 지령이다.

groups지령은 사용자를 지정하지 않으면 현재 등록가입된 사용자가 속한 그룹을 보여주며 사용자를 지정하게 되면 지정한 사용자가 속해있는 그룹들을 보여준다.

현재 사용자그룹은 admin이다.

root가 속한 그룹을 한번 살펴보자.

grpck

/usr/sbin/grpck

그룹검사

지령형식: grpck [추가선택] [파일]

-n : 삭제하겠는가를 문의하지 않는다.

/etc/group과 /etc/gshadow파일을 참조하여 그룹을 검사하는 지령이다.

grpck지령은 pwck와 류사한 지령으로서 /etc/group과 /etc/gshadow파일들을 검사하고 잘못된 부분을 찾으면 삭제하겠는가를 문의하며 "yes/no"의 형식으로 사용자의 대답을 기다린다. -n추가선택은 모든 질문의 대답을 "no"로 한다. 일반적으로 /etc/group파일이 파괴되는 경우는 별로 없다.

그룹에 새로운 계정을 추가하기 위해 /etc/group파일을 편집하던 도중에 문자를 잘못 입력했다고 가정해보자. 결국 grouptest1그룹에 root계정을 추가하려고 했는데 roo만 입력한 경우이다.

grpck지령을 실행한 결과 roo계정이 없다는 오유가 현시된다.

roo라는 계정을 지우겠는가하는 물음에 n을 입력하고 /etc/group파일의 잘못된 부분을 수정할수 있다.

grpconv

/usr/sbin/grpconv

shadow그룹생성

지령형식: grpconv

grpconv지령은 pwconv지령과 비슷한데 /etc/group파일의 그림자파일을 만들어 /etc/gshadow로 보관한다. 이것을 해제하려면 grpunconv지령을 사용한다. 그림자파일에 대한 보다 자세한 설명은 pwconv지령을 참고하기 바란다.

위험한 통과어

위험한 통과어란 추적프로그람을 리용하여 손쉽게 통과어를 알아낼수 있는 통과어를 말한다.

/etc/shadow파일을 열어보았을 때 통과어마당이 "7tqNQGFbMOk3s"인 계정이 있다고 가정하자. 이것은 원래 우리가 사용하는 암호를 crypt()함수를 사용하여 암호화한것이다. crypt()함수는 통과어를 암호화할 때 사용한 salt건을 암호화한 내용의 앞부분에 붙여 /etc/shadow파일에 저장한다. 그러나 이 함수는 거꿀알고리듬이 존재하지 않기때문에 이 암호만으로는 원래 어떤 내용이 들어있었는지 알수 없다.

현재 대부분의 추적프로그람은 이미 영어사전에 나오는 단어는 물론 사람들이 사용할만한 사람이름이나 짧은 문장 등이 포함된 사전을 리용하여 단어들을 빠른 속도로 암호화하여 통과어와 비교한다. 즉 추적프로그람은 crypt("단어", "salt건")의 형식으로 단어를 차례로 대입한다. 수자인 경우에도 일정한 범위안에서는 간단하게 해독할수 있다.

gunzip

/usr/bin/gunzip

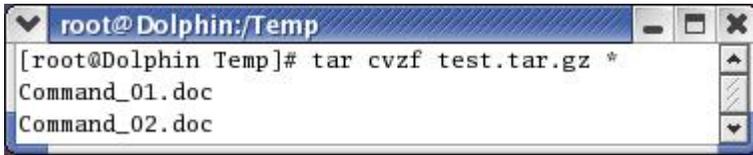
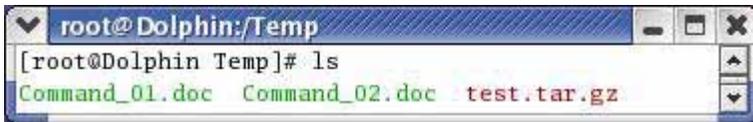
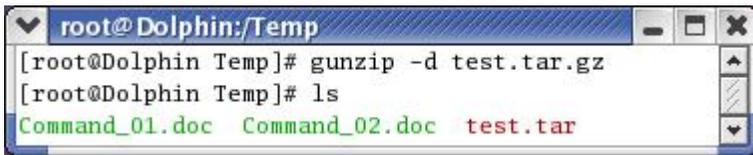
압축해제

지령형식: gunzip [추가선택] [파일]

-f, --force : 파일에 다중련결이 있거나 다른 말단에서 사용하는 경우 또는 파일이 이미 존재하는 등 여러가지 문제가 생기더라도 압축을 강제로 해제한다.
-h, --help : 도움말을 보여준다.
-l, --list : 압축된 파일의 크기, 압축해제가 될 때의 크기, 압축률, 압축이 해제될 때의 파일이름과 같은 정보들을 보여준다.
-L, --license : gunzip의 허가정보를 보여준다.
-f, --recursive : 등록부를 압축한 경우 그 등록부에 속한 파일까지 압축을 해제한다.

-t, --test : 파일이 제대로 압축되였는가를 검사한다.
-v, --verbose : 진행상태를 자세하게 보여준다.
-V, --version : 판본정보를 보여준다.

gzip형식으로 압축된 파일을 해제하는 지령이다. 이것은 gzip -d와 같은 역할을 한다.

먼저 등록부의 파일들을 하나의 파일로 압축해보자. tar cvzf지령은 현재등록부에 있는 파일을 *.tar파일로 묶은 다음 그 파일을 *.tar.gz라는 파일이름으로 압축한다.

ls지령으로 현재등록부에 있는 파일들이 test.tar.gz로 압축된것을 볼수 있다.

.gz확장자를 가진 압축파일은 gunzip지령으로 해제할수 있다.

혹은 gunzip지령대신에 gzip -d를 리용해도 된다.

```
# gzip -d test.tar.gz
```

파일을 압축해제하는것과 동시에 tar로 묶인 파일을 해제하려면 tar xvzf추가선택을 사용하면 된다.

```
# tar xvzf test.tar.gz
```

gzexe

/usr/bin/gzexe

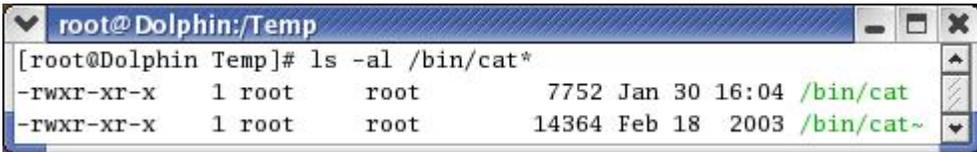
실행파일압축

지령형식: gzexe [파일 ...]

gzexe편의프로그람은 실행파일을 압축하는데 압축된 파일도 원래의 파일과 같이 실행할수 있으므로 디스크용량을 줄일수 있게 한다.

먼저 /bin/cat지령을 압축해보자.

```
# gzexe /bin/cat
```

파일을 압축하면 원래파일이름뒤에 (~)를 붙여 자동여벌파일을 만들고 압축된 파일은 원래의 파일이름으로 생성된다.

이 압축된 파일의 용량은 절반으로 줄었지만 압축된 상태에서도 이전 지령과 같이 cat의 기능을 수행할수 있다.

주의 압축된 파일의 실행은 원래파일의 실행보다 속도가 느린 경향이 있다.

gzip

/usr/bin/gzip

파일압축

지령형식: gzip [추가선택] [파일]

gzip방식의 압축률이 매우 높은 파일압축프로그람으로서 Linux에서 가장 널리 사용되는 압축방식이다.

tar지령과 함께 tar.gz로 쓰이는 경우도 많다. gzip는 파일을 압축만 할수 있고 묶지는 못한다. 여러개의 파일을 하나로 묶어 파일전송이나 관리를 쉽게 하기 위해서는 반드시 tar지령과 함께 리용해야 한다.

gzip는 압축한 후에 원본파일이 없이 .gz확장자를 가진 파일을 새로 만들어낸다. 이때 파일의 호출권한과 방식은 원래의 파일의것이 그대로 유지된다.

upgrade등록부에 있는 모든 파일들을 .gz파일로 압축해보자.

```
root@Dolphin:~
[root@Dolphin root]# ls upgrade/
4Suite-0.11.1-13.i386.rpm     apmd-3.0.2-18.i386.rpm
a2ps-4.13b-28.i386.rpm        ash-0.3.8-8.i386.rpm
acl-2.2.3-1.i386.rpm          aspell-0.33.7.1-21.i386.rpm
alchemist-1.0.26-1.i386.rpm   at-3.1.8-33.i386.rpm
anacron-2.3-25.i386.rpm       atk-1.2.0-2.i386.rpm
[root@Dolphin root]# gzip -r upgrade
[root@Dolphin root]# ls upgrade/
4Suite-0.11.1-13.i386.rpm.gz
a2ps-4.13b-28.i386.rpm.gz
acl-2.2.3-1.i386.rpm.gz
alchemist-1.0.26-1.i386.rpm.gz
anacron-2.3-25.i386.rpm.gz
apmd-3.0.2-18.i386.rpm.gz
ash-0.3.8-8.i386.rpm.gz
aspell-0.33.7.1-21.i386.rpm.gz
at-3.1.8-33.i386.rpm.gz
atk-1.2.0-2.i386.rpm.gz
```

이것은 개별적인 파일들로 압축한것이므로 tar지령을 리용하여 다시 하나의 파일로 묶을수 있다.

주의 gzip -rd를 사용해도 같은 결과를 얻을수 있다.

압축을 풀기 위해서는 다음과 같이 한다.

```
# tar xvzf upgrade.tar.gz
```

halt

/sbin/halt

체계완료

지령형식: halt [-w] [-d] [-f]

-w : 실지로는 완료하지 않고 /var/log/wtmp기록을 남긴다.
-d : wtmp파일에 기록을 남기지 않는다.
-f : 강제로 완료한다.

체계를 안전하게 끝내는 체계관련지령이다.

halt지령은 /var/log/wtmp파일에 체계완료기록을 남기고 체계완료나 재기동작업을 한다. 만약 runlevel이 0이나 6이 아닌 체계에서 halt나 reboot지령을 사용하면 shutdown(8)지령이 호출된다.(-h나 -r추가선택기능을 함)

체계완료지령

shutdown, halt, init를 리용하여 체계를 완료 혹은 재기동할수 있다.

init 0(체계완료) , init 6 (체계재기동)

지령은 여러 사용자가 체계에서 작업할 때 작업을 마무리할수 있는 완료경고통보를 보여주지 않고 완료되므로 봉사기체계에서는 사용하지 말아야 한다.

shutdown

체계완료에 대한 경고통보문을 다른 사용자들에게 보여주고 완료하는 시간을 설정하여 매 사용자가 자기의 작업을 마무리할수 있는 시간적여유를 줄수 있다.

halt
shutdown -r now와 같은 기능을 하는 지령이다.
reboot
shutdown -r now와 같은 기능을 하는 지령이다.

hdparm

/sbin/hdparm

하드디스크정보 보기/설정

지령형식: hdparm [추가선택] [장치이름]

-a [sectcount] : 하드디스크의 앞서읽기기능을 설정하거나 정보를 볼수 있다.
-A [0 또는 1] : 하드디스크의 읽기잠금설정을 취소/선택할수 있다.
-c [chipset_mode] : (E)IDE 32-bit I/O방식을 설정하거나 정보를 볼수 있다.
-C : 전원방식상태를 검사한다.
-d [0 또는 1] : DMA기능을 취소/설정할수 있다.
-E : CD-ROM의 속도를 설정한다.
-f : 장치의 리용을 중지하기 위해 고속완충기(buffer cache)를 동기화시킨다.
-g : 구동기의 물리적배치를 보여준다.
-h : 사용정보를 보여준다.
-i : 기동시간에 얻을수 있는 구동기의 정보를 볼수 있다.
-I : 구동기의 정보를 직접 볼수 있다.
-k [0 또는 1] : 재기동시에 설정내용을 보관하는 기능을 설정하거나 정보를 볼수 있다.
-K [0 또는 1] : 재기동시에 특징들을 보관하는 기능를 설정한다.
-L [0 또는 1] : 구동기에 자물쇠를 설정/해제한다.
-m [sectcount] : 구동기의 다중분구수를 설정하거나 정보를 볼수 있다.
-p [0~5중 하나] : 구동기의 미리꺼내기수를 설정한다.
-P [sectcount] : 장치의 내부적인 미리꺼내기구조를 위한 최대분구수를 설정한다.
-q : 추가선택을 화면에 표시하지 않는다. -i, -v, -t, -T에는 적용되지 않는다.
-r [0 또는 1] : 읽기전용방식으로 설정하거나 정보를 보여준다.
-R : IDE대면부를 등록한다.
-S [timeout] : 구동기의 실행대기시간을 설정한다.
-T : 고속완충기억기의 읽기시간을 볼수 있다.
-t : 장치읽기시간을 볼수 있다.
-u [0 또는 1] : 장치의 새치기마스크를 설정하거나 정보를 보여준다.
-U : IDE대면부를 제거한다.
-v : -i추가선택을 제외한 모든 설정을 보여준다.

-W [0 또는 1] : IDE구동기의 쓰기완충기능을 설정/해제한다.
-X [xfermode] : 새로운 (E)IDE/ATA2구동기를 위해 IDE전송방식을 설정한다.
-y : IDE구동기를 대기상태로 전환한다.
-Y : IDE구동기를 일시중지상태로 전환한다.
-Z : 전원자동보호기능을 해제한다.

하드디스크, CD-ROM 등의 장치설정을 보고 조종하는 지령이다.
하드디스크의 정보를 보도록 하자. 16-bit I/O를 지원하고 DMA기능을 사용하고있다.

```
root@Dolphin:~
[root@Dolphin root]# hdparm /dev/hda4

/dev/hda4:
 multcount    = 16 (on)
 IO_support   =  0 (default 16-bit)
 unmaskirq    =  0 (off)
 using_dma    =  1 (on)
 keepsettings =  0 (off)
 readonly     =  0 (off)
 readahead    =  8 (on)
 geometry     = 4865/255/63, sectors = 10266480, start = 6123600
```

CD-ROM장치의 DMA기능은 using_dma로 설정하는데 on으로 설정되여있다.

```
root@Dolphin:~
[root@Dolphin root]# hdparm -d /dev/hda4

/dev/hda4:
 using_dma    =  1 (on)
```

DMA기능을 off로 설정하려면 -d 0추가선택을 사용한다.

```
root@Dolphin:~
[root@Dolphin root]# hdparm -d 0 /dev/hdc

/dev/hdc:
 setting using_dma to 0 (off)
 using_dma    =  0 (off)
```

일부 Red Hat계렬의 Linux배포판이 설치된 체계에서 LG CDROM 40배속과 48배속을 사용할 경우 CD인식이 안되거나 체계가 꺼지는 현상이 종종 발생하였다. 이것은 체계의 DMA기능을 사용할 때 생기는 문제이다. hdparm지령으로 DMA사용을 금지시키면 해결된다.

```
# hdparm -c 1 -d 0 /dev/cdrom
```

사용하지 않을 때 하드디스크 정지하기

60분이상 유휴상태가 되면 자동으로 하드디스크의 전원을 끈 상태로 만들수 있다. 다음과 같이 입력하면 된다.

```
# hdparm -S 60 /dev/hda
```

head

/usr/bin/head

파일의 앞부분 보기

지령형식: head [추가선택] [파일이름]

-행수 : 현시하려는 행수를 쓴다. 행수가 10이라면 10개의 행을 현시한다.
-c, --bytes=SIZE : 현시하려는 용량을 정할수 있다. SIZE는 수자뒤에 b(block=512bytes), k(Kilobytes), m(Megabytes)과 같은 문자들을 덧붙여 용량을 구분해주어야 한다. 뒤에 단위가 없을 경우 byte단위로 보여준다.
-n, --lines=LINES : 현시하려는 행수를 쓴다.
-q, --quiet, --silent : 현시할 때 파일이름을 보여주지 않는다.
-v, --verbose : 현시하는 파일이름을 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일의 앞부분에서부터 지정한 행까지의 내용을 내보낸다.

이 지령은 주로 긴 파일에서 앞의 일부분만을 보려고 할 때 많이 쓰이며 기록파일을 확인할 때에도 tail지령과 함께 리용된다.

이 지령을 리용하여 mail파일의 첫 10개 행을 읽어보자. -v추가선택을 사용하면 현시되는 파일이름을 먼저 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# head -v /var/mail/root
==> /var/mail/root <==
From root@localhost.localdomain  Thu Jan 27 15:39:50 2005
Return-Path: <root@localhost.localdomain>
Received: from localhost.localdomain (Dolphin [127.0.0.1])
        by localhost.localdomain (8.12.8/8.12.8) with ESMTP id j0RKdlBE003576
        for <root@localhost.localdomain>; Thu, 27 Jan 2005 15:39:49 -0500
Received: (from root@localhost)
        by localhost.localdomain (8.12.8/8.12.8/Submit) id j0RKdkQ1003573
        for root; Thu, 27 Jan 2005 15:39:46 -0500
Date: Thu, 27 Jan 2005 15:39:46 -0500
From: root <root@localhost.localdomain>
```

그러면 이번에는 12번째 행까지 읽어보자. 다음과 같이 직접 그 범위를 지정한다.

```
root@Dolphin:~
[root@Dolphin root]# head -12 /var/mail/root
From root@localhost.localdomain  Thu Jan 27 15:39:50 2005
Return-Path: <root@localhost.localdomain>
Received: from localhost.localdomain (Dolphin [127.0.0.1])
        by localhost.localdomain (8.12.8/8.12.8) with ESMTP id j0RKdlBE003576
        for <root@localhost.localdomain>; Thu, 27 Jan 2005 15:39:49 -0500
Received: (from root@localhost)
        by localhost.localdomain (8.12.8/8.12.8/Submit) id j0RKdkQ1003573
        for root; Thu, 27 Jan 2005 15:39:46 -0500
Date: Thu, 27 Jan 2005 15:39:46 -0500
From: root <root@localhost.localdomain>
Message-Id: <200501272039.j0RKdkQ1003573@localhost.localdomain>
To: root@localhost.localdomain
```

파일용량을 기준으로 하여 일정한 크기만큼 읽어들이는 방법도 있다. 이때에는 -c추가선택을 사용한다.

```
root@Dolphin:~
[root@Dolphin root]# head -v -c 100 /var/mail/root
==> /var/mail/root <==
From root@localhost.localdomain  Thu Jan 27 15:39:50 2005
Return-Path: <root@localhost.localdomain>
```

100byte의 용량만큼 출력한다.

련관지령

cat 파일을 첫행부터 마지막까지 보여준다.

tail 파일의 뒤부분을 보여준다.

tac 파일을 마지막행부터 웃방향으로 올라가면서 보여준다.

Linux가 세상에 알려진 날

1991년 8월 25일, 핀란드의 헬싱키대학에 다니던 리누스 토발즈라는 대학생이 minix뉴스그룹에 한통의 우편을 보냈다. 이 우편의 기본내용은 지난 4월부터 자체로 개발한 조작체계가 거의 완성단계에 이르렀다는것이였다.

이것이 바로 Linux가 세상에 처음으로 알려진 사건이였다. 이어 그해 9월 17일 Linux핵심부 0.01판본이 처음으로 공개되였다.

host

/usr/bin/host

콤퓨터정보 보기

지령형식: host [추가선택] [령역 / IP] [봉사기]

봉사기 : 령역이나 IP를 검색할 이름봉사기를 지정한다. 만일 지정되지 않으면 체계에 등록된 령역봉사기를 검색한다. 이것은 /etc/resolv.conf파일에 저장되여있다.
-l : 지정된 령역안에 있는 모든 콤퓨터들을 다 보여준다.
-v : 자세한 정보를 보여준다.
-w : DNS봉사기의 응답을 기다린다.
-r : 반복처리를 진행하지 않는다.
-t [type] : 류형에 따르는 정보를 본다. (A : 콤퓨터 IP주소, NS : 검색한 콤퓨터의 이름봉사기, PTR : 령역이름지적자, ANY : 모든 류형정보)
-a : "-t ANY"와 같은 기능을 한다.

지정한 령역의 IP를 알고싶거나 혹은 IP에 해당하는 령역을 알고싶을 때 사용하는 지령이다. 콤퓨터이름은 알고있지만 IP주소를 모르는 경우 혹은 그 반대의 경우에 사용한다.

host지령은 체계에 등록되여있는 DNS봉사기를 검색한다. 만약 다른 DNS봉사기를 검색하려면 콤퓨터이름/IP주소뒤에 봉사기의 주소를 써준다.

www.hana.co.kp의 IP주소검색에 이 령역의 DNS를 리용해보자. 그러면 사용하는 DNS봉사기의 정보와 검색한 IP주소가 출력된다.

검색하려는 류형을 -t추가선택으로 정해주면 hana.co.kp의 이름봉사기령역을 알수 있다.

DNS (Domain Name System) 봉사기

IP주소라고 부르는 수자로 된 Internet주소는 일반사람들은 리해하기 힘들기때문에 쉽게 기억하고 사용할수 있는 령역이름이 필요하였다.

매 콤퓨터체계는 IP주소로 구별하여야 하므로 IP주소와 령역이름을 같이 리용할수 있는 체계가 필요하였다. 이렇게 콤퓨터이름과 IP주소를 서로 변환하는 중계자역할을 하는것이 DNS봉사기이다. DNS봉사기는 분산형자료기지로서 현재의 DNS봉사기에서 해당한 주소를 찾지 못하는 경우에는 상위DNS봉사기에 접속하여 찾는다. 다시말하여 DNS봉사기는 망상에서 콤퓨터와 콤퓨터를 련결하여 쉽게 통신하기 위한 도구인 동시에 광범한 자료기지이기도 하다.

hostid

/usr/bin/hosted

콤퓨터ID지정

지령형식: hostid [추가선택]

--help : 간단한 도움말을 보여준다.
--version : 판본정보를 보여준다.

hostid지령은 콤퓨터ID번호를 16진수로 지정하거나 보여주는 지령이다. 이것은 콤퓨터의 고유번호로서 다른 콤퓨터들과 구별할 때와 Internet주소를 부여할 때 사용된다.

이 값은 체계관리자가 새롭게 정할수도 있으며 /etc/hostid파일에 보관되여있으므로 필요한 경우에 읽을수 있다.

콤퓨터ID와 망ID

콤퓨터ID와 망ID는 모두 IP를 구성하는 요소이다.

같은 망에 포함되여있는 체계는 같은 망ID를 사용한다. 또한 Internet에서 이 망ID는 반드시 유일해야 한다.

콤퓨터ID(혹은 콤퓨터주소)는 망에서 작업기, 봉사기, 경로기 및 TCP/IP콤퓨터를 구분하기 위하여 리용한다. 매 콤퓨터의 주소는 반드시 망ID에 대해 유효해야 하며 같은 망안에서는 유일해야 한다.

hostname

/bin/hostname

체계이름보기

지령형식: hostname [추가선택] 파일이름

-a, --alias : 콤퓨터의 별명을 보여준다.
-d, --domain : 령역이름을 보여준다.
-F, --file 파일이름 : 파일로부터 콤퓨터이름을 받아서 설정한다.
-f, --fqdn, --long : 긴 콤퓨터이름을 보여준다.
-h, --help : 도움말을 보여준다.
-i, --ip : 콤퓨터의 IP주소를 보여준다.
-s, --short : 짧은 콤퓨터이름을 보여준다.
-V, --version : 판본정보를 보여준다.
-v, --verbose : 콤퓨터이름을 설정하거나 콤퓨터이름을 자세히 보여준다.
-y, --yp, --nis : NIS령역이름을 설정하거나 보여준다.

체계이름을 확인하고 설정하는 지령이다.
추가선택이 없이 이 지령을 사용하면 FQDN형태로 출력된다.

콤퓨터이름을 변경한다.

FQDN (Fully Qualified Domain Name)

Linux설치때 망설정과정에서 령역이름을 입력하는 대화칸이 나온다. 령역이름을 입력할 때 www.secbook.co.kp가 령역이름인지 secbook.co.kp가 령역이름인지 명백하지 않으므로 FQDN을 사용한다. 사실 콤퓨터이름은 앞에 나오는 www이고 령역이름은 secbook.com이다. 그리고 www.secbook.co.kp는 FQDN이라는 용어로 정의한다. 이것은 령역이름과 주콤퓨터이름을 함께 묶어 사용한다는것을 의미한다.

www	secbook.com	www.secbook.com
콤퓨터이름	령역이름	FQDN

IPv6

32bit IPv4로 Internet를 사용하는 사람들의 수가 폭발적으로 증가하는것으로 하여 현재 사용하는 TCP/IP주소구조로서는 그 많은 콤퓨터들에 주소를 다 할당할수 없게 되였다. 그래서 나온것이 IPv6이다. IPv6은 128bit의 주소구조를 가진 TCP/IP의 다음세대표준이다. IPv6은 32bit의 주소정보를 가지는 IPv4보다 4배나 많은 정보를 가질수 있으므로 주소할당의 수요를 충분히 보장할수 있게 되였다.

IPv6은 모든 규격이 Internet표준을 정의하는 IETF(Internet Engineering Task Force)의 요구대로 설계되였다.

id

/usr/bin/id

계정ID확인

지령형식: id [추가선택] [사용자이름]

-g, --group : 사용자의 그룹ID만 보여준다.
-G, --groups : 추가그룹의 ID만 보여준다.
-n, --name : -u, -g, -G추가선택과 같이 쓰이며 해당한 ID의 이름만을 보여준다.
-r, --real : -u, -g, -G추가선택과 같이 쓰이며 해당한 실제 ID를 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

사용자의 UID, GID번호 및 현재쉘에서의 ID정보를 보여준다.

id지령은 현재사용자의 ID와 유효한 사용자ID, 그룹ID를 보여준다. 내부 bash 변수인 $UID, $EUID, $GROUPS와 대응된다. 사용자를 지정하면 해당한 사용자의 정보를 보여준다.

id지령으로 현재 등록가입한 계정에 대한 정보를 보자.

주의 EUID에 대한 설명은 chmod부분을 참고하기 바란다.

우의 실행화면을 통하여 현재 등록가입된 사용자는 root계정이고 uid는 0이며 gid는 0이라는것을 알수 있다.

이러한 정보는 echo $UID지령으로도 알아볼수 있다.

ifconfig

/sbin/ifconfig

망대면부설정

지령형식: ifconfig [대면부]
ifconfig 대면부 [류형] 추가선택 | 주소

대면부 : 대면부이름. NIC가 설정되여있다면 eth0, eth1로 되며 PPP로 련결되여있다면 ppp0, ppp1로 된다.
류형 : 지정한 대면부에서 사용할 규약을 지정한다. 현재 지원되는것으로는 inet(TCP/IP를 사용할 경우), inet6(IPv6), ax25(AMPR Packtet Radio), ipx(Novell IPX) 등이 있다.
up : 지정한 대면부를 능동으로 한다.
down : 지정한 대면부를 비능동으로 한다.
[-]arp : ARP규약을 능동/비능동으로 한다.
[-]promisc : 무차별방식을 능동/비능동으로 한다. 무차별방식을 능동으로 하면 대면부를 통과하는 모든 파케트를 받는다.
[-]allmulti : 다중수신자송신(all-multicast)방식을 능동/비능동으로 한다. 이 방식을 능동으로 하면 대면부를 통과하는 모든 파케트를 받는다.
metric N : metric대면부를 설정한다.
mtu N : MTU대면부를 설정한다.
dstaddr addr : PPP 원격IP주소를 설정한다.
netmask addr : 대면부의 망마스크를 설정한다.
add addr/prefxlen : 대면부에 IPv6주소를 할당한다.
del addr/prefxlen : 대면부에서 IPv6주소를 제거한다.
irq addr : 장치에 irq주소를 지정한다.
io_addr addr : 장치의 입출구주소를 지정한다.
mem_start addr : 장치의 공유기억기의 시작주소를 지정한다.
media type : 장치의 물리적형태를 설정한다. 물리적형태에는 10base2(thin Ethernet), 10baseT(twisted-pair 10Mbps Ethernet), AUI(external transceiver) 등이 있다.
[-]broadcast [addr] : 대면부의 발신주소를 설정한다.
[-]pointtopoint [addr] : 대면부의 점대점방식을 능동으로 하고 주소를 설정한다.
hw class address : 대면부의 하드웨어장치주소를 설정한다.
multicast : 대면부를 다중송수신상태로 설정한다.
address : 대면부에 IP주소를 설정한다.
txqueuelen length : 장치의 전송대기렬의 길이를 설정한다.

Ethernet망기판과 망환경을 설정하는 지령이다.
먼저 -a추가선택으로 설정되여있는 대면부를 살펴보자.

```
[root@Dolphin root]# ifconfig -a
eth0  Link encap:Ethernet  HWaddr 00:0C:76:00:DD:52
      inet addr:192.168.8.83  Bcast:192.168.8.255  Mask:255.255.255.0
      UP BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:12 errors:0 dropped:0 overruns:0 frame:0
      TX packets:106 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:702 (702.0 b)  TX bytes:4452 (4.3 Kb)
      Interrupt:5 Base address:0x1000 Memory:fc500000-fc500038

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:4324 errors:0 dropped:0 overruns:0 frame:0
      TX packets:4324 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:310378 (303.1 Kb)  TX bytes:310378 (303.1 Kb)
```

IP설정

eth0대면부의 망IP와 망마스크를 아래와 같이 설정해보자.

```
# ifconfig eth0 192.168.8.83 netmask 255.255.255.0
```

설정된 eth0의 대면부를 살펴보자.

```
[root@dolphin root]# ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:0C:76:00:DD:52
      inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:146 errors:0 dropped:0 overruns:0 frame:0
      TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:14911 (14.5 Kb)  TX bytes:168 (168.0 b)
      Interrupt:5 Base address:0x1000 Memory:fc500000-fc500038
```

대면부를 능동으로 설정

eth0대면부를 능동으로 한다.

```
# ifconfig eth0 up
```

아래와 같이 ifup지령으로도 같은 결과를 얻을수 있다.

```
# ifup eth0
```

대면부를 비능동으로 설정

eth0대면부를 비능동으로 하기 위해서는 다음과 같이 입력한다.

```
# ifconfig eth0 down
```

주의 ifdown eth0이라고 입력해도 같은 결과를 얻을수 있다.

기본관문 설정

route지령을 리용한 기본관문설정은 다음과 같다.

```
# route add -net default gw 211.255.253.62
```

설정된 관문은 route지령으로 확인할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# route
Kernel IP routing table
Destination     Gateway          Genmask          Flags Metric Ref    Use Iface
192.168.8.0     *                255.255.255.0    U     0      0        0 eth0
169.254.0.0     *                255.255.0.0      U     0      0        0 eth0
127.0.0.0       *                255.0.0.0        U     0      0        0 lo
default         sec.linuxroot.c  0.0.0.0          UG    0      0        0 eth0
```

IP alias 설정하기

eth0대면부의 별명으로 eth0:0을 설정해보자.

```
root@dolphin:~
[root@dolphin root]# ifconfig eth0:0 192.168.1.1 netmask 255.255.255.0
[root@dolphin root]# ifconfig eth0:0
eth0:0    Link encap:Ethernet  HWaddr 00:0C:76:00:DD:52
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:5 Base address:0x1000 Memory:fc500000-fc500038
```

PROMISC방식의 대면부를 설정 및 확인

promisc방식에서는 내부망의 파케트정보를 살펴볼수 있다.

```
root@dolphin:~
[root@dolphin root]# ifconfig eth0 promisc
[root@dolphin root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:76:00:DD:52
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:264 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:25784 (25.1 Kb)  TX bytes:168 (168.0 b)
          Interrupt:5 Base address:0x1000 Memory:fc500000-fc500038
```

promisc방식으로 설정된 대면부를 아래와 같이 해제한다.

```
root@dolphin:~
[root@dolphin root]# ifconfig eth0 -promisc
[root@dolphin root]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:76:00:DD:52
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:268 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:26054 (25.4 Kb)  TX bytes:168 (168.0 b)
          Interrupt:5 Base address:0x1000 Memory:fc500000-fc500038
```

관문(gateway)

관문은 망으로 들어가는 입구역할을 수행한다. 경로선택의 시점에서 보면 Internet는 많은 관문과 콤퓨터들로 구성된 망이라고 볼수 있는데 망사용자의 콤퓨터와 웨브페지, 파일에 대한 봉사를 제공하는 봉사기는 주콤퓨터이며 망에서 통행을 관리하거나 Internet봉사를 제공하는 콤퓨터는 관문에 속한다. 관문의 역할을 하는 콤퓨터가 대리봉사기나 방화벽봉사기의 역할을 함께 수행하는 경우도 종종 있다.

info

/usr/bin/info

초본문지도서

지령형식: info [추가선택] 항목

--directory path : info문서를 찾을수 있는 등록부경로가 추가된다. 기정경로는 /usr/info 혹은 /usr/local/info이다.
-f filename, --file filename : 지정한 파일을 보여준다.
-n node, --node node : 지정한 마디에 해당한 파일을 보여준다.
-o filename, --output filename : 결과를 화면에 보여주지 않고 지정한 파일에 복사한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

info지령은 조작체계 및 련관된 프로그람들에 대한 설명문을 보여준다. 이 info파일은 대부분 /usr/local/info등록부 혹은 /usr/info등록부에 있다. 이외에 사용자지도서를 보는 지령으로서는 man지령이 있다.

이 지령이 실행되면 다음과 같은 지령을 사용할수 있다.

기본지령

h	info지령의 사용법을 보여준다.
?	info에 대한 지령요약을 보여준다.
Ctrl-g	작업진행중지, 지령취소
Ctrl-l	화면을 새롭게 갱신한다.

다른 마디로 이동하기

n	다음 마디로(next) 이동한다.
p	이전 마디로(previous) 이동한다.
u	웃마디로(up) 이동한다.
m	일정한 마디를 지정하여 그 마디로 이동한다.
f	호상참조사용. 이 건을 누르면 호상 참조할 이름을 물어본다. 이때 m사용법과 같이 사용하면 된다.
l	마지막 마디로 이동한다.

마디안에서 이동하기

[space]	한 화면 아래로 이동.
[del]	한 화면 우로 이동.
b	첫 마디로 이동.

고급지령

q	info를 완료한다.
1	마디안에 있는 차림표의 첫번째 항목으로 이동한다. 수자를 지정하게 되면 지정한 항목으로 이동한다.
g	지정한 마디로 이동한다.
s	문자렬을 찾기 위해 아래에 문자렬입력칸을 현시한다. 현재 마디에서 지정한 문자렬을 찾으면 그곳으로 유표를 이동한다.

man지령에 비해 리용률은 적지만 매우 편리하므로 점점 사용자가 늘어나고있다.

init

/sbin/init

초기화프로세스의 조종

지령형식: /sbin/init [-t sec] [0123456SsQq]

0,1,2,3,4,5,6: 실행할 준위를 나타낸다. 프로세스들의 준위는 /etc/inittab에서 확인할수 있다.
Q, q: /etc/inittab파일을 다시 읽는다.
S, s: /sbin/init를 단일사용자방식으로 전환한다.
-t 시간(s) : init가 TERM신호를 보낸후 지정한 시간(s)만큼 기다린 후 KILL신호를 보낸다.
　초기값은 5s이다.

root@dolphin:~

```
[root@dolphin root]# cat /etc/inittab
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

기동시에 실행준위에 따라 여러 프로세스를 호출하여 실행하는 지령이다.

핵심부가 기억기에 적재되면 뒤이어 init지령이 실행된다. init지령은 여러가지 프로세스를 실행하고 프로세스를 시작하는 스크립트를 호출한다.

우선 /etc/inittab를 읽어 체계의 실행준위를 결정하고 등록가입을 위한 getty 프로그람을 실행시키며 여러가지 데몬을 배경방식으로 실행시킨다.

아래에 /etc/inittab파일을 보여준다.

실행준위는 Linux계렬마다 다를수 있다. 이 지령의 실행준위를 초기값으로 5로 지정하였는데 5는 X11로 실행하면 된다. 만일 조작탁방식으로 기동하고 싶다면 id:5:initdefault로 변경하면 된다.

Run gettys in standard runlevels라는 주석아래의 mingetty는 등록가입을 관리하는 getty형태의 지령이다. 실행준위 5로 기동한 후 [Ctrl+Alt+F1]부터 [Ctrl+Alt+F6]을 입력하면 개별적인 조작탁의 등록가입화면을 볼수 있다. 만일 조작탁의 화면을 줄이려면 요구하는 수자만 남겨두고 나머지는 모두 삭제하거나 주석처리(#)한다. inittab파일을 수정한 다음에는 init -q지령으로 수정된 내용을 적용할수 있다.

insmod

/sbin/insmod

핵심부모듈 불러들이기

지령형식: insmod [추가선택] 모듈파일 [symbol=value]

-f : 핵심부판본이 다르더라도 강제로 모듈을 적재한다.
-h : 도움말을 보여준다.
-k : 모듈을 자동으로 삭제할수 있게 한다.
-L : 같은 모듈이 동시에 적재되지 못하게 한다.
-m : 적재배치정보를 생성한다.
-n : 모듈을 적재하지 않고 보여주기만 한다.
-p : 모듈이 핵심부와 정합되는가를 확인한다.
-s : syslog파일에 오유내용을 기록한다.
-v : 오유를 구체적으로 보여준다.
-V : 판본정보를 보여준다.
-o NAME : 모듈을 지정한 이름으로 설정한다.
-p prefix : 핵심부나 모듈을 미리 정의한다.

사용할수 있는 핵심부모듈을 적재하여 동작하게 하는 지령이다.

적재할수 있는 모듈들은 /lib/modules/핵심부판본/kernel/등록부에 있다. 매

하위등록부에서 모듈파일들을 살펴보자.

```
root@dolphin:/lib/modules/2.4.20-8/kernel
[root@dolphin root]# cd /lib/modules/2.4.20-8/kernel
[root@dolphin kernel]# ls
arch  crypto  drivers  fs  lib  net
```

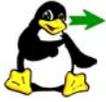
lsmod지령으로 현재 기억기에 적재되여있는 ip_tables모듈, 음향관련모듈, Ethernet망기판모듈, ext3모듈, usb관련모듈 등을 볼수 있다.

```
root@dolphin:/lib/modules/2.4.20-8/kernel
[root@dolphin kernel]# lsmod
Module                  Size  Used by    Not tainted
nls_iso8859-1           3516   1  (autoclean)
nls_cp437               5116   1  (autoclean)
vfat                   13004   1  (autoclean)
fat                    38808   0  (autoclean) [vfat]
ide-cd                 35708   1  (autoclean)
cdrom                  33728   0  (autoclean) [ide-cd]
i810_audio             27720   0  (autoclean)
ac97_codec             13640   0  (autoclean) [i810_audio]
soundcore               6404   2  (autoclean) [i810_audio]
i830                   74336   1
agpgart                47776  11  (autoclean)
parport_pc             19076   1  (autoclean)
lp                      8996   0  (autoclean)
parport                37056   1  (autoclean) [parport_pc lp]
autofs                 13268   0  (autoclean) (unused)
e100                   60644   1
ipt_REJECT              3928   6  (autoclean)
iptable_filter          2412   1  (autoclean)
ip_tables              15096   2  [ipt_REJECT iptable_filter]
keybdev                 2944   0  (unused)
```

우의 모듈중에서 ip_tables모듈을 rmmod지령으로 기억기에서 제거해보자. 제거된 ip_tables모듈은 modprove지령이나 insmod지령으로 다시 적재할수 있다.

```
# rmmod ip_tables
```

```
root@dolphin:/lib/modules/2.4.20-8/kernel
[root@dolphin kernel]# insmod ip_tables
Using /lib/modules/2.4.20-8/kernel/net/ipv4/netfilter/ip_tables.o
```

이렇게 insmod와 rmmod 그리고 lsmod지령으로 모듈들을 자유롭게 관리할수 있다.

련관지령

depmod 적재할 핵심부모듈의 의존성을 다루는 지령
modprobe 핵심부모듈을 적재하는 지령

핵심부웨브봉사기(httpd) 사용하기

핵심부웨브봉사기의 가속기데몬(kHTTPd)은 핵심부에 내장된 웨브봉사기로서 모듈형태로 사용할수 있다. 파일체계의 파일만을 봉사하기때문에 cgi와 같은 동적내용은 다룰수 없다. 그러나 Apache 등 일반 웨브봉사기보다 빠르게 봉사할수 있으므로 다른 웨브봉사기와 련동하여 사용하면 좋다. 즉 일반파일봉사는 kHTTPd가, cgi 등은 Apache가 처리하도록 한다면 좋은 효과를 낼수 있다.

이것은 아직 완벽한 프로그람이 아닌데다가 웨브봉사기로 사용하지 않는 봉사기에서 동작하면 보안상 문제가 생길수 있으므로 모듈로 선택하게 되여있다.

install

/usr/bin/install

파일설치

지령형식: install[추가선택] [파일]등록부

-d, --directory : 목적등록부를 지정하며 만일 없으면 등록부를 만든다.
-g group, --group group : 지정한 그룹ID나 그룹이름으로 새로운 파일의 그룹을 설정한다.
-m mode, --mode mode : 지정한 방식값으로 새로운 파일의 속성을 설정한다. 기정값은 0755이다.
-o [owner], --owner [owner] : 소유자를 지정한다. 만일 지정하지 않으면 root로 한다.
-s, --strip : 기호표를 제거한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

지정한 속성으로 정해진 위치에 파일을 설치하는 지령이다.

cp지령과 비슷하지만 필요하면 파일이 복사될 경로를 만들고 다른 소유자나 그룹을 지정하는 등의 여러가지 기능을 가지고있으므로 콤파일된 프로그람을 정확한 위치에 설치할 때 많이 사용된다. 특히 Makefile에서 많이 사용한다.

현재등록부의 file_change_or_mv.sh스크립트실행파일을 /bin등록부에 설치하자.

```
# install -o admin -g admin file_change_or_mv.sh /bin/
```

ipchains

/sbin/ipchains

방화기능설정

지령형식: ipchains 지령 [추가선택]

-A chain, --append chain : chain을 추가한다.
-c chain, --check chain : 설정한 chain규칙을 검사한다.
-D chain, --delete chain : chain규칙을 삭제한다.
-D chain 규칙번호, --delete chain 규칙번호 : chain규칙에서 지정한 규칙번호를 삭제한다. 만일 규칙번호가 1이라면 chain규칙의 첫번째 규칙을 삭제한다.
-F [chain], --flush [chain] : 모든 chain규칙을 삭제하거나 지정한 chain규칙을 삭제한다.
-I chain [규칙번호], --insert chain 규칙번호 : chain규칙을 지정한 번호위치에 삽입하거나 마지막에 삽입한다.
-L [chain], --list [chain] : 모든 chain규칙을 보거나 지정한 chain규칙을 본다.
-ML, --masquerading -list : 현재 가면(Masquerading)접속된 상황을 본다.
-Ms tcp tcpfin udp, --masquerading --set tcp tcpfin udp : 가면시간초과값을 설정한다.
-N chain, --new-chain chain : 새로운 chain규칙을 만든다.
-p chain target --policy chain target : chain규칙을 지정한 chain규칙으로 교체한다.
-R chain 규칙번호, --replace chain 규칙번호 : chain규칙가운데서 지정한 번호위치의 규칙을 교체한다.
-z [chain], --zero [chain] : 모든 chain규칙을 0으로 만들거나 지정한 chain규칙을 0으로 만든다.
-b, --bidirectional : 두가지 규칙을 삽입한다. (-s와 -d)
-p [!] proto, --protocol [!] proto : 규약을 지정한다. [!]는 제외한다는 의미이다.
-s [!] address[/mask] [!] [port[:port]], --source [!] address[/maskl [!] [port[:port]] : 출발지주소를 지정한다. 포구이름이나 포구번호도 지정할수 있다.
[!] [port[:port]] : 출발지의 포구이름이나 포구번호만을 지정할수 있다.
-d [!] address[/mask] [!] [port[:port]], --destination [!] address[/mask] [!] [port[:port]] : 목적지주소를 지정한다. 마찬가지로 포구이름이나 포구번호를 지정한다.
[!] [port[:port]] : 목적지의 포구이름이나 포구번호만을 지정할수 있다.
[!] typename : tcp/udp/icmp의 형태를 지정할수 있다.

-i [!] name[+], --interface [!] name[+] : 망대면부를 지정할수 있다. name+는 name이 공통으로 들어가는 모든것을 말한다.
-j target [port], --jump target [port] : 지정하는 target나 포구로 재지정한다.
-m [+-]mark : 정합되는 파케트에 수자를 매긴다.
-n, --numeric : IP주소와 포구번호를 수자그대로 보여준다.
-l, --log : 일치되는 파케트를 syslog에 남긴다.
-v, --verbose : 상세한 정보를 보여준다.
-x, --exact : 정확한 값으로 보여준다.
--line-numbers : 규칙번호만 보여준다.
--no-warnings : 경고통보문을 비능동으로 한다.
[!] -y, --syn : TCP파케트에서 SYN만을 설정한다.
-V, --version : 패키지의 판본정보를 보여준다.

ipchains는 파케트려과형태의 방화벽기능을 관리하는 도구이다. 파케트려과는 방화벽실현의 가장 기본적인 형태로서 파케트머리부의 정보를 보고 파케트의 통과(ACCEPT), 무시(DENY/REJECT)와 같은 역할을 수행한다.

먼저 설정되여있는 chain규칙이 있는가를 확인한다. -L은 chain규칙을 보여준다.

```
[root@Dolphin root]# ipchains -L
Chain input (policy ACCEPT):
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
```

Linux에서는 일반적으로 input, forward, output를 chain규칙이름의 기정값으로 규정한다. 우에서 보는바와 같이 chain규칙이 설정된것이 없다.

input chain규칙에 시작주소가 192.168.0.1인 규칙을 주도록 하자.

```
# ipchains -A input -s 192.168.0.1
```

정확히 되였는가를 확인해본다.

```
[root@Dolphin root]# ipchains -L
Chain input (policy ACCEPT):
target    prot opt     source       destination   ports
-         all  ------  192.168.0.1  anywhere      n/a
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
```

prot는 procotol을 말하며 source는 192.168.0.1, destination은 anywhere로서 0/0과 같다. ports는 설정되지 않았다.

반대로 192.168.0.1을 제외하는 규칙을 주자면 -s ! 192.168.0.1로 한다. 이것은 포구나 규약에서도 사용할수 있다.

이번에는 설정된 규칙을 -D chain규칙번호로 삭제해보자. 여기서 규칙번호는 input chain규칙에서 입력된 순서로서 -L input로 표시되는 순서로 생각하면 된다.

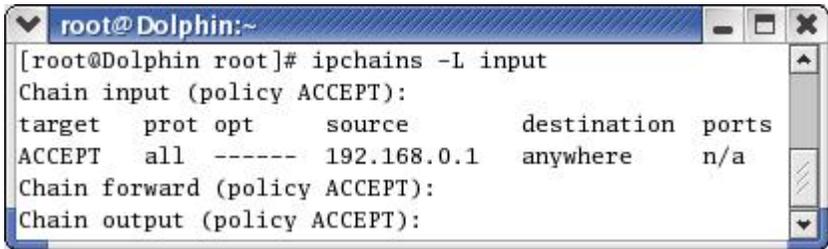
```
# ipchains -D input 1
```

-L input추가선택으로 input chain규칙을 다시 보면 정확히 삭제된것을 알수 있다.

input chain규칙에 출발지가 192.168.0.1인 파케트를 받는 규칙을 추가해보자.

```
# ipchains -A input -s 192.168.0.1 -j ACCEPT
```

다시 확인해보자.

```
root@Dolphin:~
[root@Dolphin root]# ipchains -L input
Chain input (policy ACCEPT):
target    prot opt      source         destination   ports
ACCEPT    all  ------   192.168.0.1    anywhere      n/a
Chain forward (policy ACCEPT):
Chain output (policy ACCEPT):
```

우와는 달리 target가 ACCEPT로서 파케트에 대한 허용을 의미한다. 만일 target가 DENY라면 파케트가 없어진다. REJECT도 파케트가 없어지지만 그 결과는 출발지주소에 알려진다.

-s추가선택은 192.168.8.83으로 출발지주소를 지정하였다. 만일 목적지주소를 지정하자면 -d address추가선택으로 address에 주소를 입력해야 한다.

-F추가선택을 사용하면 모든 chain규칙은 삭제된다.

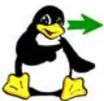
간단한 실례를 통하여 살펴보자.

우선 input chain규칙만을 사용하였고 1, 2, 3번째에서 Samba(139)와 ssh(22), 웨브봉사(80)는 syn파케트에 대해 모든 IP에서 접근할수 있도록 허용하였다.

4,5번째에서 lo와 eth0의 대면부접근을 허용하였다. 6,7번째에서 이름봉사를 받기 위해 61.40.233.124, 168.126.63.1의 목적지주소를 허용하였다. 그중에서 이름봉사는 UDP규약만을 사용하므로 -p추가선택으로 udp파케트만을 허용하도록 설정하였다.

8,9번째에서 나머지 출발지주소와 목적지주소로 오는 TCP규약을 가진 파케트는 모두 DENY로 하고 UDP규약은 REJECT하는데 그래도 접근하는 파케트는 -l추가선택으로 syslog에 남기게 된다.

모든 ipchains규칙을 입력하면 기억기에 적재되므로 콤퓨터를 재기동하면 모두 없어진다. 이것을 ipchains-save지령으로 보관할수 있다.

보관한 규칙파일은 ipchains-restore지령으로 회복할수도 있다.

련관지령

ipchains-restore　보관한 규칙스크립트를 복귀한다.
ipchains-save　지령행에서 진행한 규칙을 보관한다.
ipfwadm　핵심부 2.0에서의 핵심부도구이다.
iptables　핵심부 2.0에서의 핵심부도구이다.

가면(masquerading) **(ipchains)**

가면수법으로 한 콤퓨터에 두개의 망대면부를 설정한 다음 하나는 외부의 Internet와 련결하고 다른 하나는 내부관문(192.168.0.1이라고 가정)의 역할을 하도록 설정한다. 내부관문의 역할을 하는 대면부와 련결된 하브를 통하여 같은 192.168.0.0/24 대역의 IP를 설정하여 내부에서도 Internet를 사용할수 있도록 할수 있다. 아래의 스크립트는 간단한 가면수법을 리용하여 내부에서 Internet를 공유할수 있게 한다.

```
/sbin/ipchains -F
echo "1" > /proc/sys/net/ipv4/ip_forward
```

ipfwadm

/sbin/ipfwadm

핵심부 2.0의 방화벽관리도구

지령형식: ipfwadm -A 지령파라메터 [추가선택]
ipfwadm -I 지령파라메터 [추가선택]
ipfwadm -O 지령파라메터 [추가선택]
ipfwadm -F 지령파라메터 [추가선택]
ipfwadm -M [-l | -s] [추가선택]

-A [방향]: IP통계분석규칙. 선택적으로 방향(in, out, 또는 both)을 보여줄수 있다. 기정방향값은 both이다.
-I : IP입력방화벽규칙.
-O : IP출력방화벽규칙.
-F : IP전달방화벽규칙.
-M : IP가면관리. 이 범주는 -l 또는 -s추가선택과만 조합하여 사용할수 있다.

지령파라메터들은 수행할 동작을 보여주는데 특별한 설명이 없으면 한 지령행에서 단 한가지만을 사용할수 있다.
-a [규칙] : 선택한 목록에 대하여 하나 또는 그이상의 규칙을 추가(Append)한다.
-i [규칙] : 선택한 목록의 맨 앞에 하나 또는 그이상의 규칙을 삽입한다.

-d [규칙] : 선택한 규칙목록으로부터 하나 또는 그이상의 규칙을 삭제한다.
-l: 선택한 목록에 있는 모든 규칙을 보여준다.
-z : 선택한 목록의 모든 규칙에 대하여 파케트와 바이트계수값을 0으로 설정한다.
-f: 선택한 규칙목록을 비운다.
-p [규칙]: 선택한 방화벽류형에 대한 기본규칙을 바꾼다.
-s tcp tcpfin udp : 가면에 사용되는 초과시간값을 바꾼다.
-c : 선택한 방화벽류형에 대하여 IP파케트가 허가, 거부 또는 거절되는가를 검사한다.
-h : 도움말을 보여준다.

지령파라메터들은 추가, 삽입, 삭제, 검사 지령과 함께 사용할수 있다.
-P 규약 : 규칙 또는 점검할 파케트의 규약을 말한다.
-S 주소[/마스크] [포구 ...] : 출발지주소와 포구를 지정한다.
-D 주소[/마스크] [포구 ...] : 목적지주소와 포구를 지정한다.
-V 주소 : 파케트를 받거나 보내는 통로로 되는 대면부에 대한 주소를 지정한다.
-W 이름 : 파케트를 주고 받는 대면부의 이름을 지정한다.
-b : 쌍방향방식으로서 규칙을 쌍방향의 IP파케트에 적용하도록 한다.
-e : 확장출력. 이 추가선택을 주면 list지령에 대하여 대면부주소와 규칙추가선택(존재한다면)을 추가로 보여준다.
-k : ACK비트가 설정된 TCP파케트와만 부합한다.
-m : 전달하려는 파케트에 대하여 가면설정한다.
-n : 수값출력. IP주소와 포구번호가 수값형태로 출력된다.
-o: 부합하는 파케트에 대하여 핵심부를 잠근다.
-r [포구] : 파케트를 지역적소케트로 재설정한다.
-t AND마스크 XOR마스크 : IP머리부의 TOS마당을 변경하기 위해 사용하는 마스크이다.
-v : 자세한 내용을 출력한다.
-x : 수자를 확장한다.
-y : SYN비트가 설정되여있고 ACK비트가 없는 TCP파케트와만 부합한다.

핵심부 2.0의 IP방화벽 및 통계분석관리도구이다.

iptables

/sbin/iptables

핵심부 2.4의 방화벽관리도구

지령형식: iptables -[ADC] chain 세부규칙 [추가선택]
iptables -[RI] chain 규칙번호 세부규칙 [추가선택]
iptables -D chain 규칙번호 [추가선택]
iptables -[LFZ] [chain] [추가선택]
iptables -[NX] [chain]
iptables -E 이전 chain이름 새로운 chain이름
iptables -P chain target [추가선택]
iptables -h

-A chain, --append chain : chain을 추가한다.
-D chain, --delete chain : chain에서 규칙을 삭제한다.
-D chain 규칙번호, --delete chain 규칙번호 : chain규칙중에서 지정한 번호의 규칙을 삭제한다. 만일 규칙번호가 1이라면 chain규칙의 첫번째 규칙을 삭제한다.
-I chain [규칙번호], --insert chain [규칙번호] : chain규칙에 지정한 위치나 마지막에 규칙을 삽입한다.
-R chain 규칙번호, --replace : chain규칙들가운데서 지정한 위치의 규칙을 교체한다.
-L [chain], --list [chain] : 모든 chain규칙 또는 지정한 chain규칙을 본다
-F [chain], --flush : 모든 chain규칙 또는 지정한 chain규칙을 삭제한다.
-Z [chain], --zero : 모든 chain규칙 또는 지정한 chain규칙을 0으로 만든다.
-C chain, --check chain : 설정한 chain규칙을 검사한다.
-N chain, --new-chain : 새로운 규칙을 만든다
-X [chain], --delete-chain : 사용자가 만든 chain 또는 모든 chain을 삭제한다.
-P chain target, --policy chain target : chain규칙을 지정한 chain규칙으로 바꾼다.
-E old-chain new-chain, --rename-chain old-chain new-chain : chain이름을 바꾼다.
-p, --procotol [!] proto : 규약을 지정한다. !는 제외라는 뜻이다.
-s, --source [!] address[/mask] : 출발지주소를 지정한다. mask는 C클라스이면 255.255.255.0 또는 24bit로 표현된다.
-d, --destination [!] address[/mask] : 목적지주소를 지정한다.
-i, --in-interface [!] input name[+] : 수신하는 망대면부이름을 지정한다. name+는 name으로 시작하는 모든 대면부이름이다.
-j, --jump target : 지정한 target로 재지정한다.
-m : 지정한 macth로 확장할수 있다.
-n : IP주소와 포구번호를 수자그대로 보여준다.
-o, --out-interface [!] output name[+] : 송신하는 망대면부의 이름을 지정한다.

-v, --verbose : 자세한 정보를 보여준다.
--line-numbers : 규칙을 보여줄 때 행번호도 나타낸다.
-x, --exact : 정확한 값으로 보여준다.
-V, --version : 판본정보를 보여준다.

핵심부 2.4에서 방화벽, 규칙설정, IP가면을 조정하는 핵심부도구이다.

ipchains와 사용법이 거의 같으나 가장 큰 차이점은 확장성에 있다. 일반적으로 핵심부확장은 핵심부모듈등록부(/lib/modules/핵심부판본/kernel/net)에 존재하는데 iptables는 요구에 따라 적재된다. 그러므로 이 모듈들을 직접 적재할 필요는 없다. iptables의 확장들은 공유서고형태로 보통 /usr/local/lib/iptables에 위치한다. 배포판은 이것을 /lib/iptables나 /usr/lib/iptables에 넣는다.

ipchain이 iptables에 비해 변경된 내용들은 다음과 같다.

- 이미 만들어진 chain이름(input, output, forward)이 소문자로부터 대문자로 바뀌였다.
- -i지시자는 들어오는 대면부만을 의미하며 INPUT와 FORWARD chain에서만 동작한다. FORWORD나 OUTPUT chain은 -o로 사용한다.
- TCP와 UDP포구는 --source-port, --sport(--destination-port, --dport)추가선택으로 사용하게 된다. -p tcp 또는 -p udp추가선택과 함께 사용되여야 한다.
- TCP -y지시자는 --syn으로 바뀌였고 -p tcp다음에 와야 한다.
- DENY target는 DROP로 바뀌였다.

먼저 lsmod지령으로 ipchains모듈이 있는가를 확인한다.

```
[root@Dolphin root]# lsmod
Module                  Size  Used by    Not tainted
ipchains               51592  27
udf                    98400   0  (autoclean)
ide-cd                 35708   0  (autoclean)
cdrom                  33728   0  (autoclean) [ide-cd]
agpgart                47776   5  (autoclean)
parport_pc             19076   1  (autoclean)
lp                      8996   0  (autoclean)
parport                37056   1  (autoclean) [parport_pc lp]
autofs                 13268   0  (autoclean) (unused)
e100                   60644   1
```

ipchains모듈이 있으므로 이것을 제거하고 ip_tables모듈을 적재하자.

다음의 실례를 살펴보자.

```
# iptables -A INPUT -s 0/0 -d 0/0 --dport 139 -p tcp --syn -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 --dport 22 -p tcp --syn -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 --dport 80 -p tcp --syn -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 -i lo -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 -i eth0 -j ACCEPT
# iptables -A INPUT -s 61.40.233.124 53 -d 0/0 -p udp -j ACCEPT
# iptables -A INPUT -s 168.126.63.1 53 -d 0/0 -p udp -j ACCEPT
# iptables -A INPUT -s 0/0 -d 0/0 -p tcp --syn -j DROP
# iptables -A INPUT -s 0/0 -d 0/0 -p udp -j REJECT
```

우선 우에서는 input chain규칙만을 사용하였다.

1, 2, 3번째에서 Samba(139)와 ssh(22), 웨브봉사(80)는 syn파케트에 대해 모든 IP에서 접근할수 있도록 허용하였다.

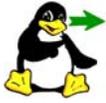
4, 5번째에서 lo와 eth0의 대면부접근을 허용하였다. 6,7번째에서 이름봉사를 받기 위해 61.40.233.124, 168.126.63.1의 목적지주소를 허용하였다. 그중에서 이름봉사는 UDP규약만을 사용하므로 -p추가선택으로 udp파케트만을 허용하도록 설정하였다.

8, 9번째에서 나머지 출발지주소와 목적지주소로 오는 TCP규약을 가진 파케트는 모두 DROP로 하고 UDP규약은 REJECT로 한다.

우에서 iptables규칙을 모두 입력하면 기억기에 적재되므로 콤퓨터를 재기동하면 없어진다. 이것을 iptables-save지령으로 보관할수 있다.

```
# iptables-save > iptables_test
```

아래와 같이 iptables-restore지령으로 저장한 규칙파일을 회복할수도 있다.

```
# iptables-restore < iptables_test
```

련관지령

iptables-save　지령행에서 설정한 iptables규칙을 보관한다.
iptables-restore　보관한 iptables규칙을 회복시킨다.

Solaris

Solaris는 BSD계렬의 SunOS를 바탕으로 하고있는 OS이다.

System V계렬의 영향을 받은 SunOS 5부터는 Solaris 2.x라는 이름도 함께 사용해왔다. 그후 Solaris 2.7부터는 다시 Solaris 7이라는 이름으로도 사용되고있다.

ispell

/usr/bin/ispell

철자검사

지령형식: ispell [추가선택] [파일]

-b : 파일을 여벌복사한다. (.bak)
-d 사전파일 : 체계표준사전외에 지정한 사전을 리용한다.(.dic)
-n : nroff/troff파일을 검사한다.
-t : TeX, LaTeX파일을 검사한다.
-x : 여벌복사하지 않는다.
-h, --help : 도움말을 보여준다.
-v : 판본정보를 보여준다.

주어진 본문파일에서 철자가 잘못 입력된 부분을 표시하는 지령이다.

파일의 철자를 검사하여 체계사전에 수록된 내용과 맞지 않는 경우 수정할수 있는 단어를 목록으로 만들어 선택할수 있게 한다.

영어만 확인할수 있다.

리용할수 있는 ispell지령을 다음의 표에 보여주었다.

지 령 어	내 용	지 령 어	내 용
i) Ignore	변경안함	I) Ignore all	모두 변경안함
r) Replace	변경	R) Replace all	모두 변경
a) Add	사전에 단어 추가	x) Exit	ispell완료

"Have you had your brekfast?"라는 문장이 포함되여있는 ispelltest파일을 검사한다.

우에서 보는바와 같이 철자가 틀린 단어가 반전표시되고 아래에는 수정할수 있는 단어목록을 보여주어 사용자가 선택할수 있게 한다. 수정하려는 단어의 번호를 선택하면 단어가 수정된다.

TeX

TeX는 일종의 조판프로그람이며 WYSIWYG방식을 지원하지 않는 단어처리기라고 볼수도 있다.

마치 html을 사용하는것처럼 하나하나 태그를 써넣으면서 사용하므로 불편하기는 하지만 긴 문장을 편리하고 론리적으로 작성할수 있으며 다양한 문자나 수식, 그림 등을 마음대로 삽입할수 있고 문서양식을 쉽게 결정할수 있으며 자주 사용하는 태그를 간단히 줄인 형태로 정의할수 있는 등 여러가지 기능들을 지원한다. TeX는 사용하기 불편하므로 보다 편리한 LaTeX 또는 수식편집에 편리한 AMSTeX 등을 사용하기도 한다. TeX는 문서의 형식을 통일지울수 있으며 수학, 물리학, 공학 등에서 사용하는 수식을 명백하게 나타낼수 있고 콤파일하지 않은 원천상태에서도 용량이 매우 작으므로 론문작성에 많이 사용되고있다. 현재 Linux문서들은 거의 TeX를 사용하여 만들어졌다.

join

/usr/bin/join

마당단위로 파일합치기

지령형식: join [추가선택]... 파일1 파일2

-a 파일번호 : 지정한 파일을 기준에 맞게 합친다.
-e EMPTY : 빈 마당을 지정하며 공백을 나타낸다.
-i, --ignore-case : 비교대상마당이 다를 경우 무시한다.
-j FIELD : -1 FIELD -2 FIELD와 같다.
-j1 FIELD : -1 FIELD와 같다.
-j2 FIELD : -2 FIELD와 같다.
-o FORMAT : FORMAT형식에 따라 합쳐진 결과를 보여준다.
-t CHAR : 지정한 CHAR를 구분자로 리용한다.
-v 파일번호 : 지정한 파일을 기준으로 비교결과를 보여준다.
-1 FIELD : 첫번째 파일의 지정한 마당번호를 기준으로 결합하여 보여준다.
-2 FIELD : 두번째 파일의 지정한 마당번호를 기준으로 결합하여 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

join은 두 파일을 일정한 규칙에 따라 하나의 파일로 만드는 지령으로서 같은 마당이 들어있는 행들을 합친다.

실례로 jointest파일과 jointest2파일이 있다고 하자.

다음의 실행화면은 첫번째와 두번째 파일의 첫번째 마당을 기준으로 같은 내용을 조사하여 보여준다.

마당기준을 -j1이나 -j2추가선택으로 변경할수 있다. -j2 1추가선택은 두번째 파일의 첫번째 마당을 기준으로 결합하는 추가선택이다.

-v추가선택은 주어진 2개 파일을 비교하여 지정된 마당에 대하여 공통으로 들어있지 않는 내용을 보여준다.

kbd_mode

/usr/bin/kbd_mode

건반방식

지령형식: kbd_mode [추가선택]

-S, --scancode : scancode방식으로 지정 (RAW)
-k, --keycode : keycode방식으로 지정 (MEDIUMRAW)
-a, --8bit : ASCII방식으로 지정 (XLATE)
-u, --unicode : UTF-8방식으로 지정 (UNICODE)
--mode={8bit, keycode, unicode, scancode} : 방식을 지정한다.
-h, --help : 도움말을 보여준다.
-V, --version : 판본정보를 보여준다.

현재 건반의 방식을 확인하고 설정할수 있다.

```
[root@Dolphin root]# kbd_mode
The keyboard is in raw (scancode) mode
```

RAW방식으로 설정되여있음을 알수 있다. 설정된 내용을 변경하기 위해서는 우의 추가선택을 참고로 입력하면 된다.

kbdrate

/sbin/kbdrate

건반속도설정

지령형식: kbdrate [추가선택]

-v : 판본정보를 보여준다.
-s : kbdrate를 출력하지 않고 실행한다.
-r rate : 건반의 rate값을 변경한다. rate는 1s동안 입력할수 있는 문자수를 말한다. (rate의 범위 : 2.0~30.0)
-d delay : 건반의 delay값을 설정한다. delay시간은 사용자가 하나의 건을 누르고있을 때 이 입력이 련속된 문자의 입력인가 하는것을 판단하는 시간이다. delay시간은 1/1000s단위를 사용한다. (건입력지연시간 : 250 ~ 1000ms)

건반의 입력속도를 조절하여 타자속도가 빠른 사람의 타자실수를 줄인다.

```
[root@Dolphin root]# kbdrate
Typematic Rate set to 10.9 cps (delay = 250 ms)
```

건으로 1s동안에 입력할수 있는 문자수를 cps(Character Per Second)의 단위로 표시한다. 즉 s당 10.9개의 문자를 입력할수 있다. 또한 250ms가 지나면 그 건을 련속 누르는것으로 보고 현재 누르고있는 건을 계속 입력하게 된다. rate 10.9cps와 delay 250ms는 IBM계렬 콤퓨터에서 기본적으로 사용되는 표준이다.

kill /bin/kill

프로쎄스완료

 지령형식: kill [-s 신호][-a] pid ...
kill -l [신호]

pid ... : 완료시킬 프로쎄스ID나 프로쎄스이름을 지정한다.
-s : 특별히 보낼 신호를 지정한다. 여기에는 신호이름이나 번호를 쓴다.
-l : 신호로 사용할수 있는 신호이름들을 보여준다. 이것은 /usr/include/linux/signal.h 파일에서도 찾아볼수 있다.
-1 : -HUP프로쎄스를 다시 능동으로 한다.
-9 : 프로쎄스를 강제로 완료시킨다.

kill지령은 지정한 프로쎄스에 지정한 신호를 보낸다. 체계에 문제가 생겨 해당 프로쎄스를 말단에서 완료하여야 할 경우에 리용한다. 만일 kill지령으로 완료되지 않는 프로쎄스가 있다면 -9추가선택으로 강제완료시킬수 있다

먼저 ps지령으로 sshd프로쎄스를 확인한다. root권한으로 pid 2754와 4454를 확인할수 있다.

```
[root@Dolphin root]# ps aux | grep sshd
root  2754  0.0  0.0  3504    4 ?      S  16:04  0:00 /usr/sbin/sshd
root  4454  0.0  0.2  3568  624 pts/0  S  16:36  0:00 grep sshd
```

참고로 ps aux지령실행시 나타나는 프로쎄스정보의 매 마당내용을 표로 보여주었다. 보다 구체적인 정보에 대해서는 ps지령을 참고하기 바란다.

root	USER	프로쎄스의 사용자
2754	pid	프로쎄스ID
0.0	%CPU	마지막 1min동안 프로쎄스가 사용한 CPU사용률
0.7	%MEM	마지막 1min동안 프로쎄스가 사용한 기억기의 사용률
3504	VSZ	가상기억기에 있는 프로쎄스의 크기(KB단위)
4	RSS	프로쎄스의 실제기억기의 크기(KB단위)
?	TTY	련결되여있는 말단
s	STAT	실행되고있는 프로쎄스상태
16:04	START	프로쎄스가 시작된 날자
0:00	TIME	프로쎄스가 소비한 총시간
sshd	COMMAND	사용자가 실행한 지령이름

ps지령으로 확인한 프로쎄스정보는 kill pid지령으로 해당 프로쎄스를 완료시키는데 리용할수 있다.

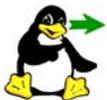
```
# kill 2754
```

만일 kill지령으로 완료되지 않는다면 -9추가선택으로 강제완료시킨다.

```
# kill -9 2754
```

kill -HUP pid지령으로 프로쎄스를 완료하였다가 다시 되살릴수 있다.

```
# kill -HUP 2754
```

련관지령

killall   프로쎄스이름으로 프로쎄스를 완료하는 지령

신호(Signal)

신호는 UNIX체계에서 프로쎄스사이에 통신하는 가장 오래된 방법으로서 프로쎄스에 비동기적인 사건을 전달하는데 사용된다. 이와 같은 신호는 건반새치기가 일어나거나 오유가 생겼을 때 발생한다. 또한 쉘이 자식프로쎄스에게 작업지령을 보낼 때에도 사용된다. 이러한 신호의 목록을 보기 위해서는 kill -l지령을 사용한다.

1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL
5) SIGTRAP	6) SIGABRT	7) SIGBUS	8) SIGFPE
9) SIGKILL	10) SIGUSR1	11) SIGSEGV	12) SIGUSR2
13) SIGPIPE	14) SIGALRM	15) SIGTERM	17) SIGCHLD
18) SIGCONT	19) SIGSTOP	20) SIGTSTP	21) SIGTTIN
22) SIGTTOU	23) SIGURG	24) SIGXCPU	25) SIGXFSZ
26) SIGVTALRM	27) SIGPROF	28) SIGWINCH	29) SIGIO
30) SIGPWR	31) SIGSYS	33) SIGRTMIN	

신호의 개수는 체계마다 다를수 있다. 프로쎄스는 대부분의 신호를 무시할수 있지만 19) SIGSTOP와 9) SIGKILL은 무시할수 없다. 앞에서 이 지령을 설명할 때 뒤에 붙던 수자가 바로 이 신호번호로서 대체로 9번이나 19번을 사용한다.

프로쎄스가 신호를 받으면 이 신호를 중단할수도 있고 직접 처리하거나 핵심부에 넘길수도 있다. 핵심부에로 넘어가면 이 신호는 기정으로 동작한다.

Linux는 프로쎄스의 task_struct에 저장된 정보로 신호를 표시한다. 지원할수 있는 신호의 개수는 처리기의 단어(WORD)크기의 제한을 받는다. 32bit단어를 사용하는 체계에서는 32개까지의 신호를 사용할수 있다. 처리대기중인 신호는 signal항목에, 중단된 항목은 blocked항목에 들어간다. 중단된 신호는 이 중단이 해제될 때까지 대기상태로 남는다.

killall

/usr/bin/killall

프로쎄스완료

지령형식: killall [-egiqvw] [-신호] 이름 ...

killall -l

killall -V

-e : 이름완성기능을 지원한다.
-g : 그룹을 지정하여 프로쎄스를 완료시킨다. 한 프로쎄스그룹에서 여러개의 프로쎄스들이 발견되여도 신호는 그룹을 단위로 한번만 보내진다.
-i : 프로쎄스를 완료하기전에 확인통보문을 내보낸다.
-1 : 알려진 모든 신호이름을 목록으로 보여준다.
-q : 어떤 통보문도 표시하지 않는다.

-v : 신호를 제대로 보내는 경우 자세한 정보를 보여준다.
-V : 판본정보를 출력한다.
-w : 프로쎄스가 완료될 때까지 대기한다.

killall지령은 kill지령과 달리 프로쎄스이름을 사용하여 프로쎄스를 완료한다. httpd데몬으로 확인해보자.

```
root@Dolphin:~
[root@Dolphin root]# ps aux | grep httpd
root     4553  0.7  3.5 19776 8700 ?       S   16:51  0:00 /usr/sbin/httpd
apache   4575  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
apache   4576  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
apache   4577  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
apache   4578  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
apache   4579  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
apache   4580  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
apache   4581  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
apache   4583  0.0  3.5 19800 8728 ?       S   16:51  0:00 [httpd]
root     4683  0.0  0.2  3568  624 pts/0   S   16:52  0:00 grep httpd
```

killall뒤에 httpd라고 프로쎄스이름을 입력하여 httpd데몬을 완료할수 있다.

주의 같은이름으로 여러개의 프로쎄스가 실행되는 경우 프로쎄스번호가 작은 쪽을 완료시키면 나머지 프로쎄스도 함께 완료될수 있다.

```
root@Dolphin:~
[root@Dolphin root]# killall httpd
[root@Dolphin root]# ps aux | grep httpd
root  4700  0.0  0.2  3568  624 pts/0  S  16:56   0:00 grep httpd
```

httpd데몬을 다시 실행해보자.

```
root@Dolphin:~
[root@Dolphin root]# /etc/rc.d/init.d/httpd start
Starting httpd:
                                                  [  OK  ]
```

httpd데몬을 확인한 결과 apache소유임을 알수 있다.

```
[root@Dolphin root]# ps aux | grep httpd
root     4850  0.0  3.3 19780 8376 ?      S  17:01  0:01 /usr/sbin/httpd
apache   4853  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
apache   4854  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
apache   4855  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
apache   4856  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
apache   4857  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
apache   4858  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
apache   4859  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
apache   4860  0.0  3.3 19804 8388 ?      S  17:01  0:00 [httpd]
root     4884  0.0  0.2  3576  628 pts/0  S  17:06  0:00 grep httpd
```

killall -e추가선택으로 프로세스련관그룹을 지정하여 프로세스를 완료할수도 있다.

```
# killall -e apache
```

련관지령

killpg　　프로세스그룹에 신호를 보내는 함수

kill　　프로세스를 완료하는 지령

killall5　　자기의 실행준위가 아닌 모든 프로세스에 신호를 보내여 모든 프로세스를 완료한다.

ksyms

/sbin/ksyms

핵심부기호 보기

지령형식: ksyms [추가선택]

-a : 모든 기호를 보여준다.
-h : 렬머리부를 보여주지 않는다.
-m : 모듈정보를 보여준다.

ksyms는 핵심부기호와 모듈에 대한 정보를 정리하여 보여주는 지령이다.
핵심부기호는 /proc/ksyms파일이나 ksyms지령을 리용하여 볼수 있다.
핵심부기호가운데서 모듈에 대한 정보를 보자.

last

/usr/bin/last

등록가입시간 보기

 지령형식: last [추가선택] [사용자이름] [tty..]

-num, -n num : 지정한 수만큼의 행을 보여준다.
-f file : 지정한 파일에서 정보를 불러들인다.
-R : 보여주는 목록에서 콤퓨터이름마당은 보여주지 않는다
-a : 보여주는 목록에서 콤퓨터이름마당을 마지막에 보여준다.
-d : 다른 콤퓨터에서 접속한것만 보여준다.
-x : 체계를 완료한 기록이나 실행준위가 바뀐 상태를 보여준다.

사용자의 등록가입기록과 재기동기록을 보여주는 지령이다.

/var/log/wtmp파일은 체계를 설치했을 때부터 지금까지의 모든 사용자의 등록가입과 등록탈퇴에 대한 정보를 가지고있다. 어떤 체계에서 어떤 사용자가 등록가입하였고 등록탈퇴하였는가를 구체적으로 알수 있으므로 체계관리에 매우 유용한 정보로 된다. 이 파일의 내용을 보여주는 지령이 last지령이다.

last -num추가선택을 사용하면 last표의 뒤에서부터 지정한 수만큼 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# last -8
root     pts/0        :0.0         Mon Jan 31 17:17   still logged in
root     :0                        Mon Jan 31 17:16   still logged in
reboot   system boot  2.4.20-8     Mon Jan 31 17:14          (00:16)
root     pts/0        :0.0         Mon Jan 31 16:09 - down   (01:04)
root     :0                        Mon Jan 31 16:08 - down   (01:05)
root     :0                        Mon Jan 31 16:07 - 16:08  (00:00)
root     :0                        Mon Jan 31 16:06 - 16:07  (00:00)
reboot   system boot  2.4.20-8     Mon Jan 31 16:04          (01:09)

wtmp begins Mon Jan 31 16:04:06 2005
```

-R추가선택은 콤퓨터이름마당을 제외한 나머지를 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# last -8 -R
root     pts/0        Mon Jan 31 17:17   still logged in
root     :0           Mon Jan 31 17:16   still logged in
reboot   system boot  Mon Jan 31 17:14          (00:21)
root     pts/0        Mon Jan 31 16:09 - down   (01:04)
root     :0           Mon Jan 31 16:08 - down   (01:05)
root     :0           Mon Jan 31 16:07 - 16:08  (00:00)
root     :0           Mon Jan 31 16:06 - 16:07  (00:00)
reboot   system boot  Mon Jan 31 16:04          (01:09)

wtmp begins Mon Jan 31 16:04:06 2005
```

매 기록파일들은 어느 정도 용량이 차거나 시간이 지나면 새로운 기록파일을 생성하므로 이전의 기록들을 검사하기 위해서는 파일을 직접 지정하여야 한다.

-f추가선택은 기록파일을 열어서 확인하는데 사용한다.

Red Hat Linux에서 기록파일은 /var/log등록부에 보관된다.

체계를 재기동한 내용만 보고싶다면 grep지령을 리용하여 검색한 결과를 출력한다.

```
root@Dolphin:~
[root@Dolphin root]# last | grep reboot
reboot   system boot  2.4.20-8   Mon Jan 31 17:14  (00:30)
reboot   system boot  2.4.20-8   Mon Jan 31 16:04  (01:09)
```

첫 행을 살펴보면 체계의 핵심부판본과 재기동한 시간을 보여준다. "00:30"형식은 체계를 재기동하기전까지 사용자가 리용한 시간을 의미한다.

lastog

/usr/bin/lastlog

접속기록보기

지령형식: lastlog [추가선택]

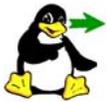
-u user : 지정한 등록가입이름을 가진 lastlog정보만을 보여준다.
-t days : 지정한 날자기간에 등록가입한 정보만을 보여준다.

lastlog지령은 /var/log/lastlog파일의 정보를 보여준다. 여기에서는 사용자들이 마지막으로 등록가입한 날자와 콤퓨터이름, 포구 등을 볼수 있다. 만일 관리자나 일반사용자를 제외한 /etc/passwd에 등록된 사용자 혹은 이상한 계정의 등록가입정보가 있다면 파괴자가 침범하여 파괴한 흔적으로 볼수 있다.

다음의 실행화면은 lastlog지령으로 등록가입정형을 보여준다. 결과 이전에 root, admin사용자만이 등록가입했다는것을 알수 있다. 다른 사용자들은 한번도 등록가입한적이 없다.

```
root@Dolphin:~
[root@Dolphin root]# lastlog
Username         Port     From             Latest
root             :0                        Sat Feb 26 21:14:23 +0900 2005
bin                                        **Never logged in**
daemon                                     **Never logged in**
adm                                        **Never logged in**
lp                                         **Never logged in**
sync                                       **Never logged in**
.........................
pvm                                        **Never logged in**
desktop                                    **Never logged in**
radvd                                      **Never logged in**
webmaster                                  **Never logged in**
admin            tty2                      Fri Feb 18 15:05:46 +0900 2005
webmin                                     **Never logged in**
newman2                                    **Never logged in**
newwoman                                   **Never logged in**
```

-u추가선택으로 지정한 사용자정보만을 출력할수 있다. 아래의 실례를 보면 root사용자가 접속한 포구와 IP, 마지막으로 접속한 시간을 보여준다.

-t추가선택으로 현재시간부터 지정한 날자까지 최근의 등록가입자만을 뽑아볼수 있다. 아래의 화면은 마지막등록가입날자가 현재부터 14일 이내인 사용자만 보여준다.

련관지령

last /var/log/wtmp파일정보를 보여주는 지령.

who /var/log/utmp파일정보를 보여주는 지령.

POSIX (Portable Operating System Interface)

POSIX는 UNIX조작체계에 기초하고있는 일련의 표준OS규약이다. 이것은 같은 규약을 따르는 조작체계들이 프로그람을 새로 작성하지 않고도 어느 정도의 호환성을 보장도록 하기 위해 만들어졌다.

POSIX의 매 표준은 POSIX다음에 소수로 표시하도록 되여있다. POSIX.1은 C언어응용프로그람대면부의 표준이며 POSIX.2는 표준쉘과 편의프로그람에 대한것이다. 그밖에도 주요 규약들이 개발되고있다.

BSD (Berkeley Software Distribution ; Berkeley Software Design)

BSD는 UC버클리회사에서 개발하여 배포한 UNIX이다. 현재까지 BSD UNIX는 인기를 끌어왔으며 많은 UNIX체계들이 이 코드를 바탕으로 하고있다. 이 계렬의 대표적인것으로는 FreeBSD, NetBSD, OpenBSD 등이 있다.

ldconfig
/sbin/ldconfig

동적련결설정

지령형식: ldconfig [추가선택] 등록부

-C cache : /etc/ld.so.cache파일대신 지정한 파일을 리용한다.
-f conf : /etc/ld.so.conf파일대신 지정한 파일을 리용한다.
-l : 수동으로 매 서고를 련결한다.
-n : 지령행에서 지정한 등록부만을 진행하며 /usr/lib, /lib, /etc/ld.so.conf파일은 진행하지 않는다.
-N : 완충기를 갱신하지 않고 단지 련결만을 갱신한다.
-p, --print-cache : 현재 완충기에 보관된 등록부와 서고목록을 보여준다.
-r root : 지정한 root를 뿌리등록부로 리용한다.
-v, --verbose : 현재판본정보와 관련된 등록부와 련결들을 구체적으로 보여준다.
-V, --version : 판본정보를 보여준다.
-X : 련결을 갱신하지 않고 단지 완충기만을 갱신한다.
-?, --help, --usage : 도움말을 보여준다.

ldconfig지령은 공유서고완충기를 재설정하는 지령이다.

서고(library)란 체계에서 프로그람을 직접 콤파일할 때 프로그람에서 공동으로 사용하는 기능들을 모아놓은 파일이다. 프로그람에서는 하나하나의 기능을 매번 만드는것이 아니라 이미 존재하는 서고의 기능을 리용한다. 이것을 동적련결프로그람(dynamic linked program)이라고 부른다. 또한 콤파일할 때 자체 서고를 사용하는것을 정적련결프로그람(statically linked program)이라고 한다.

동적으로 련결된 실행파일은 공유서고에 의존하지만 대부분은 현재서고판본에 국한되지 않는다.

새로운 판본의 서고를 설치하자면 그것을 특별한 등록부에 설치하고 ldconfig를 실행하여 공유서고완충기를 다시 설정해야 한다. ldconfig는 존재하는 파일을 조사하고 /etc/ld.so.cache의 완충파일을 설정하면서 필요한 서고를 적재하고 기호련결을 만드는 역할을 한다.

먼저 /etc/ld.so.conf파일을 살펴보자.

ld.so.conf에서 빠진 내용은 home등록부의 .bash-profile이나 .bashrc파일에 있는 LD_LIBRARY_PATH환경변수에 등록부를 추가하여 보장한다. 다음의 실례는 admin계정에서 oracle설치를 위한 LD_LIBRARY_PATH를 추가한 내용을 보여준다.

공유서고의 내용이 변경되였으면 ldconfig지령을 실행하여 공유서고완충기를 다시 만든다.

```
# ldconfig
```

ldd

/usr/bin/ldd

서고의 의존성보기

지령형식: ldd [추가선택] 파일

--version : 판본정보를 보여준다.
-v, --verbose : 관련된 구체적인 정보를 보여준다.
-d, --data-relocs : 정해진 곳에 없는 객체의 위치를 다시 배치하고 결과를 보여준다.
-r, --function-relocs : 자료객체와 함수를 다시 배치하고 객체나 함수가운데서 찾을수 없는 결과는 보여준다.
--help : 도움말을 보여준다.

공유서고의 의존성을 확인하는 지령이다. 이 지령을 리용하면 프로그람이 사용하는 공유서고관계를 알수 있다. 례를 들어 ls지령의 공유서고목록을 살펴보도록 하자.

여기서 /lib/tls/libc.so.6은 C서고이다. 다른 프로그람작성언어에서도 그 언어의 서고를 구현하기 위하여 C서고를 쓸수 있다. 그러므로 모든 프로그람에 이 서고가 다 포함될것이다. /lib/ld-linux.so.2는 다른 모든 서고를 적재한다.

less

/usr/bin/less

파일보기

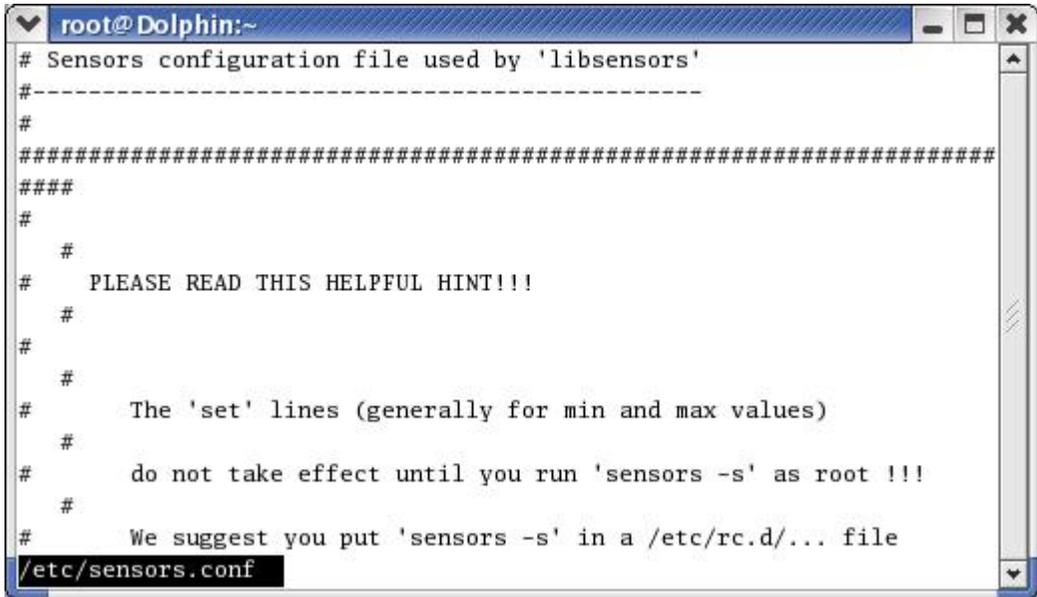
지령형식: less [추가선택] [파일이름]

-?, --help : 도움말을 보여준다.
-a, --search-skip-screen : 마지막 행이 화면에 출력된 다음 탐색을 시작한다.
-c, --clear-screen : 필요할 때 전체 화면을 우에서부터 아래로 다시 갱신한다.
-C, --CLEAR-SCREEN : -c추가선택과 같지만 갱신할 때 화면전체를 지우고 시작한다.
-e, --quit-at-eof : 파일의 끝에 두번째로 도달하면 자동적으로 완료한다. 또는 q가 입력되면 완료된다.
-E, --QUIT-AT-EOF : 파일의 끝에 도달하기만 하면 자동적으로 완료한다.
-i, --ignore-case : 검색단어자체의 대소문자를 구분하지 않고 탐색한다.
-I, --IGNORE-CASE : 검색단어에 포함된 단어의 대소문자를 구분하지 않고 탐색한다.
-m, --long-prompt : 아래행에 more지령과 같이 전체 파일내용에서 파일을 읽은 퍼센트를 표시한다.
-M, --LONG-PROMPT : 아래행에 more지령보다 상세한 정보를 보여준다.
-n, --line-numbers : 행번호를 보여주지 않는다.
-N, --LINE-NUMBERS : 매 행의 시작부분에 행번호를 현시한다.
-ofilename, --log-file=filename : 지정한 파일이름에 보여지는 내용을 저장한다.
-Ofilename, --LOG-FILE=filename : -o추가선택과 비슷하지만 파일이름이 존재하면 확인절차없이 그 파일에 내용을 덧쓰기한다.
-q, --quiet, --silent : 일정한 오유가 발생하여도 체계경고음을 울리지 않는다.
-Q, --QUIET, --SILENT : 체계경고음을 절대로 내지 않는다.

-s, --squeeze-blank-lines : 련속되는 공백행을 하나의 행으로 처리한다.
-xn, --tabs=n : 지정한 n값만큼 태브간격을 조절한다. 기정값은 8이다.
-V, --version : 판본정보를 보여준다.

less지령은 more보다 기능이 추가된 지령으로서 페지단위로 파일의 내용을 보여준다.

/etc/sensors.cf파일을 한번 살펴보자. less지령을 실행한 다음 입력되는 대부분의 지령은 vi의 기능과 매우 류사하다.

```
root@Dolphin:~
# Sensors configuration file used by 'libsensors'
#-----------------------------------------------
#
########################################################################
####
#
   #
#    PLEASE READ THIS HELPFUL HINT!!!
   #
#
   #
#       The 'set' lines (generally for min and max values)
   #
#       do not take effect until you run 'sensors -s' as root !!!
   #
#       We suggest you put 'sensors -s' in a /etc/rc.d/... file
/etc/sensors.conf
```

실행후 h를 입력하여 사용할수 있는 지령들을 보자.

페지이동	공백건, Ctrl+V, F, Ctrl+F
Z	공백건과 비슷하지만 수자를 입력하고 z를 입력하면 지정한 수자행만큼 이동한다.
한줄아래 이동	Enter건, Ctrl+N, e, Ctrl+E, j
반페지아래로 이동	d, Ctrl+D
반페지우로 이동	b, Ctrl+B
한행우로 이동	y, Ctrl+Y, Ctrl+P, k, Ctrl+K
반페지우로 이동	u, Ctrl+U
화면 새로 고침	r, Ctrl+R, Ctrl+I
완료	q

이외에도 파일을 찾기 위해서는 /이나 ?를 입력하면 된다. /은 아래로 이동하면서 찾

으며 ?는 우로 이동하면서 찾게 된다. 하나를 찾은 후 다음것을 찾으려면 n을 입력한다.

이외에도 많은 기능이 있지만 vi편집기와 매우 류사하므로 vi편집기부분을 참고하기 바란다.

GRUB (GRand Unified Bootloader)

GRUB는 Linux와 BSD계렬, GNU/Hurd는 물론 다양한 조작체계에서 사용할 수 있는 기동적재기이다. Linux에서만 설치와 설정을 할수 있었던 LILO와는 달리 우리가 알고있는 많은 체계들에서도 설치와 설정을 할수 있다.

lilo

/sbin/lilo

Linux기동적재기

지령형식: lilo [추가선택]

-v : 구체적인 정보를 보여준다.
-q : 현재참고파일의 목록. lilo는 기본적으로 /boot/map에 의해 유지되며 boot에 대한 핵심부 이름과 위치를 포함한다. 이 추가선택은 그곳에 있는 파일들의 목록을 보여준다.
-m map-file : 지정한 map-file을 사용한다.
-c config-file : 지정한 config-file에서 설정파일을 읽는다. 기정파일은 /etc/lilo.conf 이다.
-d delay : lilo기동화면에서 건입력이 없을 경우 자동적으로 기정체계로 기동한다. 이때 자동적으로 기동되는 시간을 지정한 시간으로 한다. 시간은 0.1s단위로 입력하여야 하는데 50으로 설정되였다면 5s를 말한다.
-D label : lilo기동화면에서 기정으로 선택될 표식을 지정한다.
-r root-directory : 어떤 작업을 하기 전에 chroot로 등록부를 변경한다. 플로피디스크로 기동하여 설정을 변경할 경우에 사용한다.
-t : 새로운 기동분구나 참고파일을 생성하지 않고 검사만을 수행한다.
-c : 린접한 분구로부터의 요청을 하나로 한다.
-f disk-tab : 지정한 파일을 디스크구조파라메터로 한다. 기정값은 /etc/disktab이다.
-i boot-sector : 지정한 boot-sector파일로 새로운 기동분구를 생성한다. 기정값은 /boot/boot.b이다.
-l : 분구/머리부/실린더주소대신에 선형분구주소(linear sector address)를 생성한다.

-s save-file : lilo가 기동분구에 덧쓰기할 때 기정값 /boot/boot.NNNN으로 이전의 내용이 보존된다. NNNN은 장치에 의존하는데 기동분구를 보존하기 위해 지정한다. 또는 -u와 함께 지정하여 기동분구를 복구하는데 리용한다.
-S Save-file : 지정한 파일로 덧쓰기한다.
-u device-name : 기동분구에 저장된 lilo를 제거한다.
-U device-name : -u와 같이 lilo를 제거하는것은 같지만 시간지표를 검사하지 않는다.
-R command line : 다음 기동시에 사용되는 기본지령을 설정한다.
-l label : 체계가 기동된 후에 환경변수 BOOT_IMAGE의 실행핵심부의 표식을 참조할수 있도록 한다.
-V : 판본정보를 보여준다.

lilo는 처음 기동을 시작할 때 기동과정을 담당할 기동프로그람을 호출하기 위해 먼저 실행되는 지령이다. GRUB와 같은 Linux기동적재기중의 하나로서 기정파일은 /etc/lilo.conf이다.

아래에 설정파일의 실례를 보여준다.

체계의 하드디스크에 설치한 OS내용은 다음과 같다. Windows는primary Master HDD의 첫번째 구획(/dev/hda1)에 설치되여있으며 Linux는 Primary Master HDD 6번째 구획(/dev/hda6)에 설치되여있다고 가정하자.

이 파일에서 설정한 추가선택을 간단히 살펴보기로 하자.

- timeout : 기동시 LILO화면에서 건반입력이 없는 경우 자동적으로 기동되는 시간을 지정한다.
- default : 다중기동을 설정하는 경우 기정으로 선택될 이름을 지정한다. 우의 실례에서 Linux는 /dev/hda6에 설치되여있으며 Windows는 /dev/hda1 구획에 설치되여있다. label추가선택에서 지정한 이름으로 Linux를 기정으로 선정하였다. 만일 이것을 Windows로 변경하면 기정선택은 window.s로 된다.
- boot : LILO가 설치될 구획을 지정한다. /dev/hda라고 지정하면 Primary Master HDD의 MBR령역에 LILO가 설치된다.
- map : 기동프로그람을 지정한다.
- message : LILO화면프로그람을 지정한다.
- lba32 : 하드디스크의 론리블로크주소화(LBA)기능을 사용한다. 용량이 큰 하드디스크에서 사용한다.
- image : Linux핵심부프로그람을 지정한다.
- label : 이름을 지정한다.
- read-only : 읽기전용으로 적재한다.
- root : lilo가 읽어들일 기본구획을 지정한다.
- other : 다중기동으로 지정할 구획을 지정한다.

lilo가 시작되지 않을 때

LILO가 시작되지 않을 때는 MBR에 제대로 설치되지 않았거나 Linux설치시 lilo 추가선택을 선택하지 않았거나 기동분구의 위치를 알수 없는 경우가 있을수 있다.

・ L : 하드디스크에 결함이 있을 때 발생하는 통보문이다.

・ LI: Map Installer를 실행하지 않고 /boot/boot.b가 이동했을 때 발생하는 문제로서 설치시 BIOS에서의 하드디스크의 형태와 설치후 BIOS에서의 하드디스크의 형태가 다를 때 발생하는 오유이다.

・ LIL : Map파일에서 서술자표를 적재하지 못할 때 발생하며 하드디스크의 결함으로 발생하는 오유이다.

・ LIL? : 두번째 기동분구가 잘못된 주소로 적재되였을 때 발생하는 오유이다.

・ LIL-: 서술자표가 파괴되였을 때 발생하는 오유이다.

LILO나 GRUB 등의 기동적재기가 사라졌을 경우 기동CD로 복구하기

Linux CD를 CD-ROM에 넣으면 boot:라고 화면왼쪽아래에 출력된다. 이때 아래와 같이 입력한다.

```
boot: vmlinuz root=/dev/hda()
```

경고: 여기서 ()는 /(기동구획)이 있는 구획이다. boot로 구획을 나누었을 경우 /boot구획번호를 지정하고 /구획만 있는 경우에는 /구획번호를 입력한다.

기동이 완료되면 root계정으로 등록가입하고 LILO설정파일을 체계에 맞게 수정한다.

```
# vi /etc/lilo.conf
```

수정후 lilo지령을 입력하여 lilo를 갱신한다.

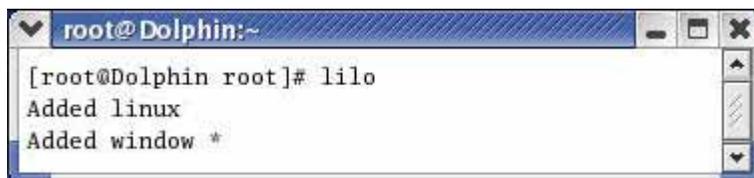
경고: 이 지령을 주지 않으면 수정한 내용이 체계에 적용되지 않는다. 핵심부를 콤파일하고 lilo.conf를 수정하여도 이 지령이 없다면 새로운 핵심부프로그람으로 기동할수 없다.

기본기동선택의 차림표 바꾸기

기본기동선택차림표를 바꾸는데는 두가지 방법이 있다.

첫번째 방법은 지령행에서 -D추가선택으로 차림표선택을 바꾸는것이다. 이때 windows는 Windows의 label이름이다.

```
# lilo -D windows
```

또 다른 방법은 /etc/lilo.conf파일에서 default추가선택을 windows의 label로 수정한 후 LILO를 갱신하는것이다.

우의 통보문에서 *가 windows로 선택된것을 볼수 있다.

디스크의 MBR령역에서 기동적재기 삭제하기

DOS상태에서 fdisk지령을 아래와 같이 실행한다.

```
# fdisk /mbr
```

ln

/bin/ln

파일련결

 지령형식: ln [추가선택] 원본 [대상]
ln [추가선택] 원본... 등록부

-b, --backup : 목적파일이 있다면 여벌복사파일을 생성한다.
-d, -F, --directory : 등록부를 하드디스크련결이 생성되게 한다. (root계정만 가능)
-f, --force : 련결을 생성할 목적파일이 있어도 강제적으로 새로운 련결를 생성한다.
-i, --interactive : 련결를 생성할 목적파일이 있을 경우 삭제하겠는가를 사용자에게 물어본다.
-n, --no-dereference : 련결할 원본이 기호파일이면 그 기호련결파일의 목적파일을 추적하여 련결한다.
-s, --symbolic : 련결할 원본이 기호련결파일이면 련결한다.
-v, --verbose : 진행되는 상태를 자세하게 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.
-S, --suffix backup-suffix : 련결을 생성할 목적파일이 이미 있을 경우 이전의 목적파일을 여벌복사할 파일의 확장자를 지정한다.
-V, --version-control {numbered, existing, simple} : 여벌복사하는 방법을 지정한다.
t, numbered : 항상 번호로 된 여벌복사파일을 만든다.
nil, existing : 목적파일이 있을 경우에만 여벌복사파일을 만든다.
never, simple : 간단한 여벌복사를 만든다.

파일을 실제경로가 아니라 사용하기 편리한 다른 경로에서 호출할수 있도록 지정하는 지령이다.

파일을 련결하는 방법에는 기호련결과 하드련결이 있다. 기호련결을 쏘프트련결

(Soft link)이라고도 한다.

기호련결, 쏘프트련결

일반적으로 련결이라고 하면 기호련결(Symbolic Link)을 말한다고 생각해도 무방하다. 기호련결은 불필요한 파일의 복사를 하지 않아도 된다. 보통 여러 등록부에서 같은 서고를 요구하는 경우와 하나의 파일을 여러 사용자가 공동으로 사용할 경우에 많이 쓰인다.

```
root@Dolphin:~
[root@Dolphin root]# ln -s /home/han/html app_data
[root@Dolphin root]# ls -al
total 208
drwxr-x---   20 root   root  4096 Jan 31 22:55 .
drwxr-xr-x   22 root   root  4096 Jan 31 22:53 ..
-rw-r--r--    1 root   root   962 Feb  1  2005 anaconda-ks.cfg
lrwxrwxrwx    1 root   root    14 Jan 31 22:55 app_data -> /home/han/html
```

하드련결

하드련결(Hard link)은 똑같은 파일크기로 원본파일이 수정될 경우 하드련결된 파일도 원본과 동일하게 변경되며 항상 같은 내용을 유지할수 있게 한다. 원본이 삭제되여도 원본과 같은 내용을 가지고있으므로 자원을 공유하면서 자료를 안전하게 관리하려고 할 때 유용하게 사용할수 있다.

기호련결과 하드련결과의 차이

	기호련결	하드련결
파일크기	작다	원본과 동일하다
원본파일 삭제시	사용할수 없다	사용할수 있다
허가권	"l"로 표시된다.	일반파일이다

하드디스크가 꽉 찼을 때

/home등록부가 꽉 찼을 때 ln을 리용하여 하드디스크의 공간을 확보할수 있다. 먼저 현재 매 구획의 남은 용량을 생각해보겠다.

사용자의 계정이 있는 구획이 꽉 차면 등록가입할수 없게 된다. 그러나 우에서 보는바와 같이 /WinData가 있는 /dev/hda5구획은 용량이 충분하다는것을 알수 있을것이다. 이때에는 사용자허가권등록부의 몇개를 /WinData밑으로 옮기고 기호련결을 걸어주는것만으로도 림시로 공간을 확보할수 있다. 물론 이렇게 하는것도 한계가 있으므로 하드디스크를 새로 얻거나 필요없는 자료를 삭제하는것이 더 효과적일것이다.

locale

/usr/bin/locale

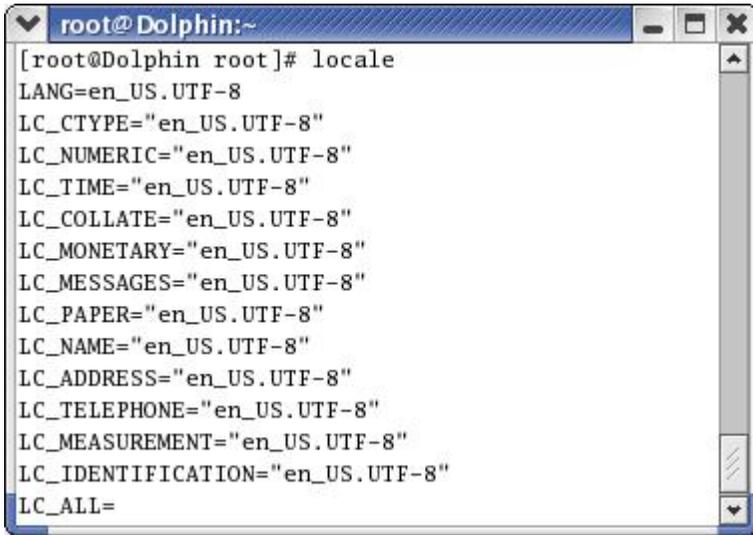
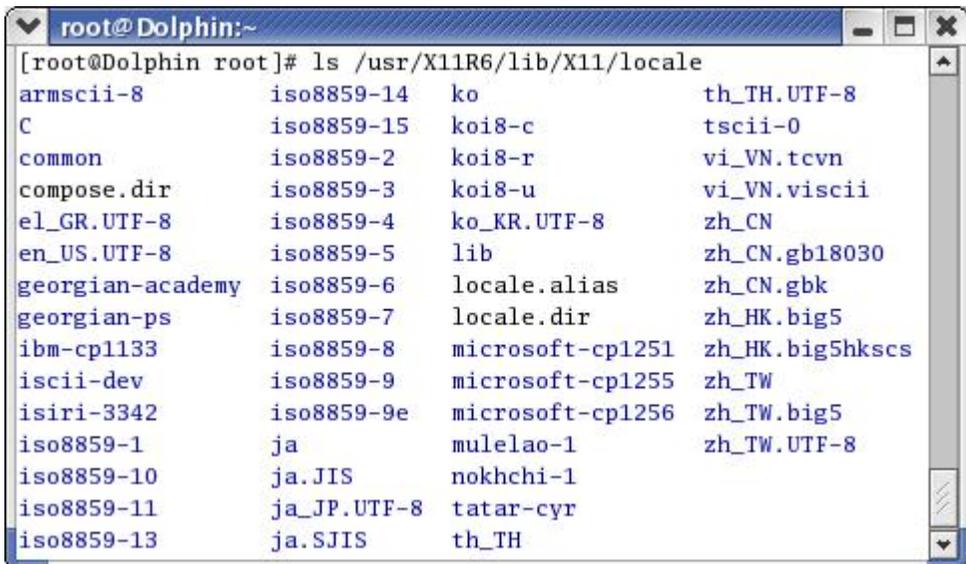
지역정보 보기

지령형식: locale [추가선택] 이름

locale [추가선택] [-a|-m]

-a, --all locales : 사용가능한 지역의 이름을 보여준다.
-m, --charmaps : 사용가능한 문자배렬의 이름들을 보여준다.
-c, --category-name : 사용가능한 범주이름을 보여준다.
-k, --keyword-name : 선택된 열쇠어의 이름을 보여준다.
-v, --verbose : 자세한 정보를 보여준다.
-?, --help : 도움말을 보여준다.
-V, --version : 판본정보를 보여준다.

locale지령으로는 매 나라별로 선택할수 있는 정보를 볼수 있다.

지역은 프로그람의 언어사항과 관련이 있다. 만일 많은 지역을 설치하였다면 다음 환경변수설정을 통하여 지역을 지원하는 프로그람의 실행방식을 선택할수 있다. 기본지역은 C 또는 POSIX로 설정되여있다. 현재체계의 지역은 locale지령으로 볼수 있다.

Red Hat계렬의 배포판에서 XWindows의 지역관련등록부는 /usr/X11R6/lib/X11/locale이다.

```
root@Dolphin:~
[root@Dolphin root]# ls /usr/X11R6/lib/X11/locale
armscii-8          iso8859-14   ko               th_TH.UTF-8
C                  iso8859-15   koi8-c           tscii-0
common             iso8859-2    koi8-r           vi_VN.tcvn
compose.dir        iso8859-3    koi8-u           vi_VN.viscii
el_GR.UTF-8        iso8859-4    ko_KR.UTF-8      zh_CN
en_US.UTF-8        iso8859-5    lib              zh_CN.gb18030
georgian-academy   iso8859-6    locale.alias     zh_CN.gbk
georgian-ps        iso8859-7    locale.dir       zh_HK.big5
ibm-cp1133         iso8859-8    microsoft-cp1251 zh_HK.big5hkscs
iscii-dev          iso8859-9    microsoft-cp1255 zh_TW
isiri-3342         iso8859-9e   microsoft-cp1256 zh_TW.big5
iso8859-1          ja           mulelao-1        zh_TW.UTF-8
iso8859-10         ja.JIS       nokhchi-1
iso8859-11         ja_JP.UTF-8  tatar-cyr
iso8859-13         ja.SJIS      th_TH
```

locate
/usr/bin/locate

파일위치찾기

지령형식: locate [추가선택]
　　　　　slocate [추가선택]

자체자료기지를 통해 파일을 찾아주는 지령이다.

locate는 현재 거의 사용되지 않으며 보안기능을 더 강화한 slocate가 많이 사용된다. 현재 사용되는 locate는 대부분이 slocate에 기호련결로 만들어진 파일이다. 자세한 내용은 slocate를 참고하기 바란다.

련관지령

fnd　　　실행지령으로 파일을 찾는 지령
slocate　　locate지령에 보안기능을 강화한 판본이다.

lockfile
/usr/bin/lockfile

잠금파일생성기

지령형식: lockfile [추가선택] 파일이름

--sleeptime : 잠금파일의 생성이 실패한 후 다시 시도하기까지의 시간을 지정한다. 기정값은 8s이다.
-r retries : 파일생성을 재시도한 때로부터 새롭게 시도할 회수를 지정한다. 기정값은 -1로써 정지하지 않고 계속 시도한다.
-l locktimeout : 잠금파일이 마지막으로 수정된 후 지정한 시간이 지나면 강제적으로 잠금파일이 제거된다.
-s suspend_time : 지정한 시간이 지난 후 잠금파일이 삭제된다. 기정값은 16s이다.
-! : 결과값의 부정이다. 주로 쉘스크립트에서 사용한다.
-ml, -mu : 우편입출력완충등록부에 허가권이 적용된다면 체계의 우편통을 잠그거나(-ml) 해제(-mu)할수 있다.

lockfile지령은 procmail패키지중의 하나로서 일정한 파일이나 장치, 자원에 대한 접근조종을 진행하는 잠금파일을 생성하는데 사용한다. 이 잠금파일은 일정한 파일, 장치, 자원을 일정한 프로세스가 사용하고있다는 기발을 리용하여 다른 프로세스의 접근을 제한한

다. 잠금파일은 /var/lock에 위치한다.

신호기발(semaphores)

기호기발은 조작체계의 자원을 경쟁적으로 사용하는 다중프로쎄스에서 처리를 조절하거나 동기화시키는 기술이다. 신호기발은 조작체계 또는 핵심부의 어떤 지정된 저장장치안에 보관되여있으며 매 프로쎄스는 이것을 확인하고 변경할수 있다. 프로쎄스는 신호기발의 값을 확인하고 그 값에 따라 즉시 자원을 사용할수 있으며 이미 다른 프로쎄스에 의해 사용중이라는 사실을 알게 되면 일정한 시간을 기다려야 한다. 신호기발은 2진수 (0 또는 1)를 사용하거나 또는 추가적인 값을 가질수도 있다. 신호기발을 사용하는 프로쎄스는 자원을 사용하는 동안에는 신호기발의 값을 변경하여 다른 사용자들이 기다리도록 해야 한다.

신호기발들은 일반적으로 기억공간을 공유하거나 파일들을 공유/호출하기 위한 두가지 목적으로 사용된다.

신호기발들은 프로쎄스사이의 통신(IPC)을 위한 기술의 하나이다. 프로그람작성언어 C에서는 신호기발을 관리하기 위한 일련의 대면부 또는 함수들을 제공한다.

Procmail

Procmail은 UNIX의 기본 우편프로그람인 Sendmail과 함께 다양하고 유용한 기능을 제공하는 우편도구이다. 그러나 설정방법이 매우 복잡하여 사용하기가 쉽지 않다. Procmail을 사용하면 손님의 전자우편을 자유롭게 조종할수 있으며 그 손님의 전자우편주소인 "xxx@령역이름"에서 "order@령역이름"으로 수신되는 전자우편을 임의의 다른 전자우편주소로 받아보게 하거나 일정한 전자우편주소로 수신되는 우편에 대한 자동응답기능을 설정할수 있는 등 여러가지 작업을 진행할수 있다.

logger

/usr/bin/logger

체계일지기록

지령형식: logger [추가선택] [통보문]

-i : 파일의 매 행마다에 일지기록의 프로세스ID를 기록한다.
-s : 체계기록과 같이 통보문을 표준오유장치에 기록한다.
-f filename : 지정한 파일의 내용을 기록한다.
-p pri : 지정한 우선권으로 통보문을 입력한다. pri로는 수자가 될수도 있고 "facility.level"형식으로 할수도 있다.
-t tag : 지정한 태그를 매 행마다 기록한다.

주어진 통보문을 /var/log/message파일에 기록한다.
그러면 아래 통보문을 기록파일에 보관시키자.

```
# logger System Logger Testing
```

/var/log/message파일의 최신기록부분에서 방금 보관한 통보문을 찾아볼수 있다.

련관지령

syslogd Linux　체계기록편의프로그람
klogd　핵심부로그데몬

login

/bin/login

체계접속

지령형식: login [추가선택] [이름]

-f user : 이미 인증절차를 거친 사용자라면 지정한 사용자의 인증절차를 무시한다. root사용자에 대해서는 적용되지 않는다.
-h hostname : 등록가입을 원격등록가입(telentd, rlogind)으로 인식하고 콤퓨터이름에 등록가입을 시도할 콤퓨터의 이름을 입력한다.
-p : 체계에 접속할 때 이전 환경들을 그대로 사용하도록 getty지령에 의해 사용된다.

체계에 접속하여 설정파일을 읽고 작업할수 있는 환경을 갖추는 인증과정지령이다.
Linux는 초기 체계를 시동할 때 init프로그람으로 설정파일을 읽어들이고 실행에 필요한 모든 getty를 생성하는데 mingetty프로그람을 사용한다.
사용자가 접속을 시도하면 mingetty는 login프로그람을 호출하며 login프로그람은 프롬프트에서 사용자이름과 암호를 읽는다. 인증절차후 입력이 제대로 되였다면 /etc/passwd에 있는 쉘을 받아 /etc/bashrc, /etc/profile과 home등록부의 .bash-rc, .bash-profile 등의 환경설정파일을 읽어들여 프롬프트를 띄운다. 쉘프롬프트에는 #와 $가 있다. #는 root쉘프롬프트이며 $는 사용자쉘프롬프트이다. 참고로 알아둘것은 사용자가 /etc/passwd파일에 접근하는것을 막는 방법에는 여러가지가 있는데 해당한 사용자설정의 맨 앞줄에 #로 주석처리를 하거나 /bin/bash를 /bin/false로 변경하면 접근을 막을수 있다. 또는 /etc/nologin파일에 사용자를 추가하여 접근을 막을수도 있다.
/etc/securetty파일에는 root로 접속할수 있는 말단들이 라렬되여있다. 이 파일이 없으면 root는 모든 말단에서 다 접속할수 있다.

logname

/usr/bin/logname

현재 사용자이름 보기

지령형식: logname [추가선택]

--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

현재 등록가입한 사용자이름을 /var/run/utmp파일에서 찾아 보여주는 지령이다.

logname과 whoami, id는 약간한 차이가 있다. logname과 whoami지령으로 현재 등록가입한 사용자를 알아낼수 있다.

id지령으로는 사용자의 UID와 GID, 속한 그룹들을 다 볼수 있다.
su지령으로 체계관리자로 등록가입한다.

이때 logname지령은 su지령으로 전환된 체계관리자가 아니라 초기에 등록가입한 사용자의 이름을 보여준다. 그러나 whoami지령은 su지령으로 전환된 root를 보여준다.

련관지령

whoami	등록가입한 사용자의 ID를 보여주는 지령
id	등록가입한 사용자의 UID, GID, groups를 보여주는 지령

logrotate

/usr/sbin/logrotate

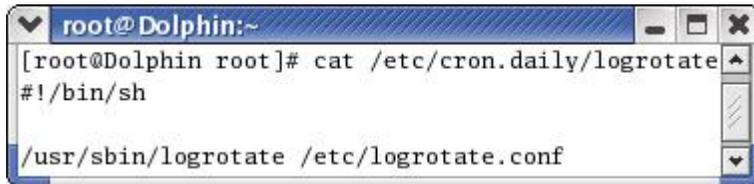
기록파일관리

지령형식: logrotate [추가선택] <설정파일>

-d, --debug : 오유수정방식.
-f, --force : 강제적으로 파일을 실행한다. 새로운 목록이 추가되면 오래된 기록파일을 수동으로 직접 삭제한 다음에 리용한다.
-s, --state [statefile] : 지정된 규칙파일을 사용하여 logrotate를 실행한다.
--usage : 간단한 도움말을 보여준다.

기록파일들을 주기적으로 압축하고 이름을 바꾸어 관리하는 지령으로서 그 설정내용은 /etc/crontab에 있다.

지정된 주기에 맞추어 매 기록파일들의 이름을 바꾸거나 압축 및 삭제한다. 보통 /etc/cron.daily등록부에 logrotate라는 스크립트를 두고 /usr/sbin/logrotate를 /etc/logrotate.conf에 보관된 설정에 맞추어서 매일 실행하게 되여있다.

아래에 /etc/cron.daily/logrotate파일의 내용을 보여준다.

/etc/cron.daily/logrotate파일은 /etc/logrotate.conf파일을 참조하여 실행된다. /etc/logrotate.conf파일의 내용을 아래에서 보여준다.

```
[root@Dolphin root]# cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this direc
include /etc/logrotate.d

# no packages own wtmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}

# system-specific logs may be also be configured here.
```

rpm패키지의 logrotate정보가 있는 등록부를 살펴보자.
매 기록파일들에는 해당한 설정내용이 저장되여있다.

우에서 보여준 파일중에서 syslog파일을 보기로 하자. syslog파일에는 messages나 secure mailog와 같은 중요한 정보가 들어있다.

```
[root@Dolphin root]# cat /etc/logrotate.d/syslog
/var/log/messages /var/log/secure /var/log/maillog /var/log/
spooler /var/log/boot.log /var/log/cron {
    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/nul
l` 2> /dev/null || true
    endscript
}
```

look

/usr/bin/look

지정한 문자렬로 시작하는 행 보여주기

 지령형식: look [추가선택] 문자렬 [파일]

-d : 영문자만을 검색한다.
-f : 대소문자를 구분하지 않는다.
-a : /usr/dict/web2파일을 사용한다.
-t character : 문자렬의 마지막 문자를 지정한다.

look지령은 파일에서 지정한 문자렬로 시작하는 행을 찾아 보여준다. 2진파일을 검색할 때에는 파일안의 매 행이 미리 정렬되여있어야 한다. 파일을 지정하지 않으면 /usr/dict/words파일을 보여준다.

/usr/bin/lpc

인쇄기관리자

지령형식: lpc [지령]

? : 도움말을 보여준다.
abort { all | printer } : 현재인쇄기데몬을 중지하고 지정한 인쇄기의 인쇄작업을 비능동으로 한다.
defaultq : LPC프로그람의 기본대기렬을 보여준다.
defaults : 설정된 기정값을 보여준다.
disable { all | printer } : 지정한 인쇄대기렬의 인쇄작업을 비능동으로 한다.
down { all | printer } : 모든 인쇄기의 대기렬과 인쇄작업을 비능동으로 하거나 지정한 인쇄기를 비능동으로 한다.
enable { all | printer } : 모든 인쇄기의 인쇄대기렬을 능동으로 하거나 지정한 인쇄기를 능동상태로 한다.
exit, quit : LPC프로그람을 완료한다.
help : 간단한 도움말을 보여준다.
lpq printer [추가선택] : lpc프로그람에서 lpq를 실행한다.
lprm printer jobid [jobid]* : lpc프로그람에서 lprm을 실행한다.
reread [printer [@host]] : LPD봉사기가 인쇄기설정정보를 다시 읽도록 한다.
start { all | printer }: 인쇄대기렬을 능동으로 하고 모든 인쇄기나 지정한 인쇄기데몬을 시작한다.
status { all | printer }: 모든 인쇄기나 지정한 인쇄기의 상태를 보여준다.
stop { all | printer }: 현재 진행중인 작업을 완료한 다음에 지정한 인쇄기데몬을 비능동으로 한다.
topq printer [jobid]: 순차적작업목록에서 지정한 작업을 인쇄대기렬의 맨 웃자리에 놓는다.
up { all | printer } : 인쇄대기렬을 능동으로 하고 지정한 인쇄기데몬을 재기동한다.

lpc는 지령행방식의 인쇄기조종프로그람이다.

이 지령으로는 인쇄기의 상태를 보거나 인쇄기 혹은 인쇄대기렬을 능동/비능동으로 하며 인쇄대기렬의 작업순서를 변경할수 있다. 보통 lpc는 한대의 콤퓨터에 여러대의 인쇄기가 련결되여있는 경우에 많이 사용된다.

우에서 보는바와 같이 "lpc>"라는 프롬프트에 지령을 입력하는 대화방식으로 인쇄기를 조종할수 있다. 또한 대화방식이 아닌 지령행에서도 가능하다.

련관지령

lpd	인쇄기데몬
lpq	인쇄대기렬의 내용을 보여준다.
lpr	파일의 내용을 인쇄한다.
lprm	인쇄대기렬의 작업을 삭제한다.
lpstat	LP인쇄기의 정보를 보여준다.

lpq

/usr/bin/lpq

인쇄대기렬 보기

지령형식: lpq [추가선택] [사용자]

-a : printcap에 있는 자료기지안에 있는 모든 인쇄기정보를 보여준다.
-l : 자세한 정보를 보여준다.

-P printer : 기정인쇄기를 사용하지 않고 지정한 인쇄기를 사용한다.
-V : 인쇄기의 판본정보를 보여준다.
num : 지정한 작업번호의 상태를 보여준다.

현재 인쇄대기중에 있는 작업내용을 보여주는 지령이다.

lpq지령은 lpd데몬을 실행한 후에 인쇄기설정이 제대로 되여있는가를 확인할 때 쓰인다. 추가선택없이 실행시키면 설정된 인쇄대기렬의 내용을 보여준다.

우에서 Rank마당은 여러개의 인쇄작업중에서 현재 진행되고있는 작업을 나타낸다. 만일 Rank가 active이면 인쇄기가 작업중이라는것을 나타낸다. Owner마당은 인쇄작업의 소유자를 나타내며 Job마당은 실행중에 있는 인쇄작업의 작업번호이다. 만일 해당한 인쇄작업을 lprm지령으로 삭제하려면 이 작업번호를 지정해야 한다. 이외에도 크기와 해당한 작업이 실행된 시간 등을 보여준다.

lpq지령을 실행하였을 때 LPD련결이 거부되는 경우가 있는데 이것은 대부분 lpd 데몬이 실행되지 않았을 때 나타나는 현상이므로 이 데몬을 실행시키면 문제를 해결할수 있을것이다.

련관지령

lpc	인쇄기관리자
lpd	인쇄기데몬
lpr	파일의 내용을 인쇄한다.
lprm	인쇄대기렬의 작업을 삭제한다.
lpstat	LP인쇄기의 정보를 보여준다.

lpr

/usr/bin/lpr

인쇄지령

지령형식: lpr [추가선택] 파일

-A : AUTH환경변수에 의해 지정된 인증을 리용한다.
-B : printcap에 지정된 려과기로 인쇄중의 파일을 려과하지 않는다.
-C class : lpq지령시 보이는 클라스를 지정한다. 이것은 A에서 Z까지의 우선권준위를 가지며 기정값은 A클라스이다. A는 우선권준위가 가장 낮으며 Z가 가장 높다.
-h : 표제띠(banner)나 머리부를 인쇄하지 않는다.
-i indentcols : 지정한 수자만큼 들여쓰기한다. 모든 인쇄기에서 다 지원하지는 않는다.
-J jobname : 작업이름을 지정한다. 지정하지 않으면 파일이름이 인쇄대기렬에서의 작업이름으로 된다.
-Kcopies, -#copies : 지정한 수만큼 파일이 반복인쇄된다.
-m mailTo : 인쇄작업이 완료되면 지정한 전자우편주소에 우편을 보낸다. 형식은 user@host이다.
-P printer : PRINTER환경변수나 체계에서 지정된 인쇄기가 아니라 지정한 인쇄기로 인쇄한다.
-r : 완충작업이 끝났을 때 파일을 제거한다.
-T title : 지정한 문자렬을 pr에 대한 파일이름대신 제목으로 사용한다.
-U username : 작업의 사용자를 지정한 이름으로 한다. 이때 root 혹은 설정파일의 allow_user_setting목록에 있는 사용자만 지정할수 있다.
-V : 진행되는 상태를 구체적으로 보여준다.
-w width : 지정한 크기로 인쇄페지의 너비를 지정한다.

lpr는 지정한 파일을 인쇄하는 지령이다.

파일을 인쇄하는 또 한가지 방법은 아래와 같이 cat지령으로 파일을 /dev/lp0에 보내는 방법이 있다. 그러나 이 방법은 root계정이나 print그룹만이 사용할수 있다.

```
# cat test.txt > /dev/lp0
```

man지령과 col -b지령을 결합하면 사용지도서의 troff로 전처리된 형식을 제거한 후 사용지도서를 보기쉽게 인쇄한다.

```
$ man man | col -b | lpr
```

-C추가선택으로는 A부터 Z까지의 우선준위클라스를 부여할수 있다. 기정값은 A클라스로서 우선권준위가 제일 낮으며 Z는 우선권준위가 제일 높다. test.txt파일을 인쇄할 때 우선준위를 B로 하려면 다음과 같이 입력한다.

```
$ lpr -C B test.txt
```

-K2나 -#2추가선택으로 해당한 파일을 지정한 수자만큼 반복인쇄할수 있다.

```
$ lpr -#2 test.txt
```

lprm

/usr/bin/lprm

인쇄취소

지령형식: lprm [추가선택] [작업번호] [사용자]

-P printer : 지정한 인쇄기에 해당되는 작업들만을 보여준다.
- : 완충기사용자가 가지고있는 모든 대기작업을 삭제한다.

lprm은 lpq지령으로 볼수 있는 작업대기렬들을 살펴보고 해당한 작업을 삭제한다. 작업번호를 지정하여 작업번호에 해당하는 인쇄대기렬을 삭제한다.

아래에서는 현재 대기중에 있는 인쇄작업들을 보여준다.

현재 인쇄중에 있는 1번 작업과 대기중에 있는 3번 작업이 있다. 이제 대기중인 3번을 삭제해보자.

```
# lprm 3
```

lpq지령으로 그 작업이 대기렬에서 사라졌는가를 확인한다.

사용자를 지정하여 해당한 사용자의 작업만을 취소할수 있는데 오직 root사용자만이 이 지령을 사용할수 있다.

```
# lprm han
```

또한 "lprm -"지령으로 완충상태에 있는 모든 작업을 취소할수도 있다.

```
# lprm -
```

련관지령

lpc　　인쇄기관리자
lpd　　인쇄기데몬
lpq　　인쇄대기렬의 내용을 보여준다.
lpr　　파일의 내용을 인쇄한다.
lpstat　　LP인쇄기의 정보를 보여준다.

ls　　/bin/ls

파일목록보기

지령형식: ls [추가선택] [파일]

-a, --all : "."을 포함한 경로안의 모든 파일을 보여준다.
-b, --escape : 도형문자가 아닌 문자들을 가진 파일들을 보여준다.
-c, --time=ctime, --time=status : 파일의 최신변경시간에 따라 파일들을 정렬시켜 보여준다. 자세한 보기(-l추가선택)이면 그 파일이 최근에 변경된 시간을 보여준다.
-d, --directory : 경로를 보여준다. (이것은 쉘스크립트에서 효과적으로 쓰인다.)
-f : 내용을 정렬하지 않고 보여준다. 즉 디스크에 보관되여있는 순서대로 보여준다. -a, -U추가선택과 같은 뜻이며 -l, -s, -t추가선택과는 반대이다.
--full-time : 시간을 구체적으로 보여준다.
-i, --inode : 파일의 왼쪽에 색인번호를 보여준다.
-k, --kilobytes : 파일크기를 보여줄 때 KB단위로 보여준다. 이 추가선택은 POSIXLY_CO RRECT 환경변수를 무시한다.
-l, --format=long, --format=verbose : 파일의 류형, 사용권한, 하드련결번호, 소유자이름, 그룹이름, 파일크기, 시간(따로 지정하지 않으면 파일이 만들어진 날자)을 구체적으로 보여준다. 현재로부터 6개월 이전의것이면 시간을 생략하고 파일의 작성년도만을 보여준다.

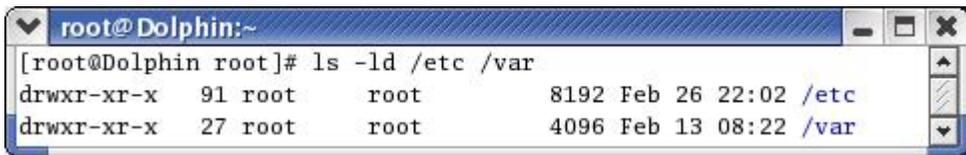
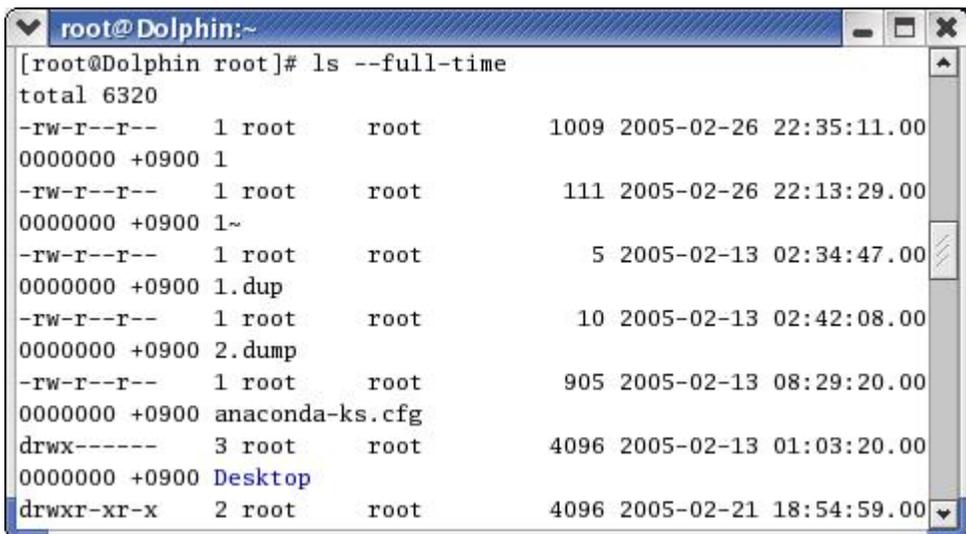
-m, --format=commas : 파일을 옆으로 정렬한다.
-n, --numeric-uid-gid : UID, GID번호를 보여준다.
-p : 파일의 형태를 가리키는 문자를 매 파일에 추가한다.
-q, --hide-control-chars : 파일의 이름에 도형문자가 아닌것이 있으면 "?"로 표시한다.
-r, --reverse : 파일들을 내림순서로 정렬한다.
-s, --size : 파일크기를 1KB단위로 나타낸다. POSIXLY_CORRECT환경변수가 지정되면 512B 단위로 지정된다.
-t, --sort=time : 시간순서로 파일을 정렬한다. 최근파일이 제일 먼저 나타난다.
-u, --time=atime, --time=access, --time=use : 파일사용시간순서로 정렬한다. 자세한 보기이면 시간표시는 만들어진 날자대신 사용된 날자를 보여준다.
-x, --format=across, --format=horizontal : 가로방식으로 정렬한다.
-A, --almost=all : ".", ".."경로를 제외한 모든 파일을 라렬한다.
-B, --ignore-backups : 파일이름의 끝이 "~"인 파일은 목록에서 제외된다.
-C, --format=vertical : 세로방식으로 정렬한다.
-F, --classify : 파일의 류형을 가리키는 문자를 매 파일뒤에 추가한다. 일반적으로 실행파일은 "*", 경로는 "/", 기호련결은 @, FIFO는 "|", 소케트는 "="을 리용하며 일반파일에는 추가되지 않는다.
-G, --no-group : 목록에서 지정한 그룹에 대한 정보를 제외한다.
-L, --dereference : 기호련결파일들을 그냥 파일로 보여준다.
-N, --literal : 이름이 영문이 아닌 경우 역사선문자(\)와 함께 출력한다.
-Q, --quote-name : -N추가선택과 반대이다.
-R, --recursive : 하위경로와 그 안에 있는 파일들도 모두 보여준다.
-S, --sort=size : 파일크기가 가장 큰것부터 정렬해서 보여준다.
-U, --sort=none : 정렬하지 않고 디스크에 보관된 순서대로 보여준다.
-X, --sort=extension : 파일의 확장자순서대로 정렬한다. 확장자가 없는 파일이 제일 먼저 정렬된다.
-1, --format=single-column : 한 행에 한개 파일씩 보여준다.
-w, --width cols : 가로길이값을 지정한다. 기정으로는 한 화면의 가로길이값이며 COLUMNS 환경변수값으로 지정할수 있다.(초기값은 80)
-T, --tabsize cols : 렬의 위치를 지정한 값으로 한다. 초기값은 8이다. 0으로 지정하면 태브문자는 무시된다.
-I, --ignore pattern : 지정된 파일들은 목록에서 제외한다.
--color, --colour, --color=yes, --colour=yes : 파일의 형태에 따라 그 파일의 색갈을 다르게 보여주는 기능을 수행한다
--color=tty, --colour=tty : --color추가선택과 같은데 표준출력에서 색갈을 사용한다. 이 추가선택은 색조종코드를 지원하지 않는 쉘스크립트나 지령행사용에서 아주 효과적으로 쓰인다.
--color=no, --colour=no : 파일을 구분하는 색을 사용하지 않는다. 이것이 초기값이다. 색사용을 이미 하고있다면 그것을 무시한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

ls지령은 지정한 등록부나 파일의 정보를 표준출력에 보여준다. 파일이나 등록부를 지정하지 않으면 현재등록부의 내용을 보여준다.

현재등록부에서 숨긴 파일까지 모두 보려면 -a추가선택을 사용한다.

```
# ls -a
```

-l추가선택은 해당되는 파일의 자세한 정보를 보여준다.

-d추가선택은 지정한 등록부의 내용만 보여준다.

--full-time추가선택을 사용하면 자세한 날자정보를 볼수 있다.

ls 지령을 사용했을 때 나오는 파일크기는 블로크단위이므로 그 크기를 알아보기 어렵다. -h추가선택은 파일의 크기를 사람이 인식하기 쉬운 단위로 보여준다.

-r추가선택은 일반적인 ls결과를 거꾸로 정렬하여 보여준다.

-t추가선택은 가장 최근에 변경된 파일들을 정렬해서 보여준다. -r추가선택을 추가하여 그 정렬순서를 반대로 할수 있다.(-trl)

-S추가선택은 파일의 크기가 제일 큰것부터 보여준다. 만일 제일 작은 파일부터 보려고 한다면 -r추가선택을 추가하여 -Srlh추가선택으로 볼수 있다.

lsattr

/usr/bin/lsattr

파일속성보기

 지령형식: lsattr [추가선택] [파일]

-R : 등록부와 그 안의 모든 파일들에 대한 속성값을 보여준다.
-a : .으로 시작하는 파일을 포함한 등록부안의 모든 파일을 보여준다.
-d : 등록부를 보여줄 때 그 안의 파일들을 보여주는것이 아니라 그 등록부를 일반파일처럼 보여준다.
-v : 파일판본정보를 출력한다.

Linux파일체계의 속성을 보여주는 지령으로서 chattr지령으로 변경한 파일의 속성내용을 볼수 있다. 구체적인 사용법은 chattr지령을 참고하기 바란다.

chattr +i추가선택으로 /etc/passwd에 i속성을 주자. i속성을 가진 파일은 지울수도 이름을 변경할수도 내용을 추가할수도 련결를 생성할수도 없게 된다.

```
# chattr +i /etc/passwd
```

chattr지령으로 변경된 속성을 보기 위하여 lsattr지령을 리용한다.

련관지령

chattr 파일속성을 변경하는 지령

lsdev

/sbin/lsdev

하드웨어보기

지령형식: lsdev

lsdev지령은 체계에 인식된 하드웨어의 장치이름과 DMA, IRQ, I/O포구 등의 정보를 보여주는 지령이다. 이 지령으로 체계에 인식된 하드웨어정보를 한눈에 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# lsdev
Device            DMA   IRQ  I/O Ports
------------------------------------------------
cascade             4     2
dma                          0080-008f
dma1                         0000-001f
dma2                         00c0-00df
e100                         1000-103f
fpu                          00f0-00ff
ICH4                      5
ide0                     14  01f0-01f7 03f6-03f6 24a0-24a7
ide1                     15  0170-0177 0376-0376 24a8-24af
Intel                        1000-103f 2000-20ff 2000-20ff 2400-243f 2
400-243f 2440-245f 2460-247f 24a0-24af
keyboard                  1  0060-006f
Mouse                    12
parport0                     0378-037a
PCI                          0cf8-0cff
pic1                         0020-003f
pic2                         00a0-00bf
rtc                       8  0070-007f
serial                       03f8-03ff
timer                     0  0040-005f
usb-uhci              10 11  2440-245f 2460-247f
vga+                         03c0-03df
```

lsmod

/sbin/lsmod

적재된 모듈 보기

지령형식: lsmod

lsmod지령으로 체계에서 현재 동작하고있는 모듈을 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# lsmod
Module                  Size  Used by    Not tainted
i810_audio             27720   0  (autoclean)
ac97_codec             13640   0  (autoclean) [i810_audio]
soundcore               6404   2  (autoclean) [i810_audio]
ip_tables              15096   0  (unused)
udf                    98400   0  (autoclean)
ide-cd                 35708   0  (autoclean)
cdrom                  33728   0  (autoclean) [ide-cd]
agpgart                47776   5  (autoclean)
parport_pc             19076   1  (autoclean)
lp                      8996   0  (autoclean)
parport                37056   1  (autoclean) [parport_pc lp]
autofs                 13268   0  (autoclean) (unused)
e100                   60644   1
microcode               4668   0  (autoclean)
nls_iso8859-1           3516   2  (autoclean)
nls_cp437               5116   2  (autoclean)
vfat                   13004   2  (autoclean)
fat                    38808   0  (autoclean) [vfat]
keybdev                 2944   0  (unused)
mousedev                5492   1
hid                    22148   0  (unused)
input                   5856   0  [keybdev mousedev hid]
usb-uhci               26348   0  (unused)
ehci-hcd               19976   0  (unused)
usbcore                78784   1  [hid usb-uhci ehci-hcd]
```

보다 구체적인 내용은 modprobe지령에서 참고하기 바란다.

련관지령

insmod　　체계에 모듈을 적재하는 지령

modprobe　보다 우월한 모듈적재지령

lsof

/usr/sbin/lsof

사용중인 파일들의 목록 보기

지령형식: lsof [추가선택] [파일]

-a : AND 연산으로 파일을 선택하여 보여준다.
-c : 지정한 COMMAND마당의 내용만 보여준다.
-F : 지정한 구분자로 마당을 구분하여 보여준다.
-g : 지정한 그룹ID를 사용하는 관련프로쎄스를 보여준다.
-i : 현재 사용되는 소케트정보를 보여준다.
-l : 등록가입한 이름대신에 UID를 출력한다.
-n : 콤퓨터이름대신에 IP주소를 출력한다.
-P : 포구봉사이름대신에 포구번호를 출력한다.
-r : 지정한 s를 주기로 반복하여 보여준다. 기정값은 15s이다.
-s : 파일의 크기를 보여준다.
-V : 자세한 정보를 보여준다.
-?, h : 도움말을 보여준다.

lsof지령은 현재 열려있는 모든 파일들에 대한 자세한 표와 매 파일에 대한 소유자, 크기, 관련프로쎄스 등의 정보를 보여준다.

COMMAND마당가운데서 sshd와 관련된 체계에서 사용하는 파일들을 보자.

```
[root@Dolphin root]# lsof -c sshd
COMMAND   PID USER   FD   TYPE DEVICE    SIZE   NODE NAME
sshd     2782 root  cwd    DIR    3,6    4096      2 /
sshd     2782 root  rtd    DIR    3,6    4096      2 /
sshd     2782 root  txt    REG    3,6  278552 392735 /usr/sbin/sshd
sshd     2782 root  mem    REG    3,6  103044 490649 /lib/ld-2.3.2.so
sshd     2782 root  mem    REG    3,6   28452 360135 /usr/lib/libwrap.s
o.0.7.6
sshd     2782 root  mem    REG    3,6   30448 490887 /lib/libpam.so.0.7
5
```

-i추가선택을 사용하여 현재 열려있는 소케트정보를 볼수 있다. TCP와 UDP를 각각 지정하여 볼수도 있다. 지금 체계에서 NAME마당의 *:봉사이름으로 현재 봉사대기상태를 볼수 있다.

```
[root@Dolphin root]# lsof -i TCP
COMMAND     PID     USER   FD   TYPE DEVICE SIZE NODE NAME
portmap    2645      rpc    4u  IPv4   2466       TCP *:sunrpc (LISTEN)
rpc.statd 2664 rpcuser    6u  IPv4   2550       TCP *:32768 (LISTEN)
sshd       2782     root    3u  IPv4   3056       TCP *:ssh (LISTEN)
xinetd     2796     root    5u  IPv4   3079       TCP Dolphin.co.kp:3276
9 (LISTEN)
sendmail   2822     root    4u  IPv4   3194       TCP Dolphin.co.kp:smtp
 (LISTEN)
spamd      2841     root    4u  IPv4   3233       TCP Dolphin.co.kp:783
```

mail

/bin/mail

우편 보내고 받기

지령형식: mail [추가선택] [-s 제목] [-c 참조] [-b 숨은참조] 받는사람
mail [추가선택] -f 파일이름
mail [추가선택] -u 사용자ID

-v : 우편을 보내는 과정의 통보문을 자세히 보여준다.
-i : tty새치기신호를 무시한다.
-n : 우편프로그람을 실행할 때 /etc/mail.rc파일을 읽지 않는다.

-N : 초기화면에서 우편목록을 보여주지 않는다.
-s 제목 : 송신우편의 제목을 적는다.
-c 참조 : 우편을 받을 사람을 함께 추가한다.
-b 숨은 참조 : 우편을 받을 사람을 함께 추가한다. 숨은 참조된 사람은 우편을 받은 사람에게는 보이지 않는다.
-f : 파일로부터 우편을 읽어온다.
-u : 사용자계정을 지정한다.

mail은 말단에서 사용하는 우편의뢰프로그람으로서 우편을 보내거나 받고 정리할수 있다.

받은 편지 확인

계정사용중에 받은 편지를 확인하려면 mail만 입력한다.

"mail -u root"지령은 관리자방식에서 일정한 계정의 사용자가 받은 편지를 확인할 때 쓰인다.

mail을 실행하면 초기화면에 받은 편지의 목록이 나온다. 초기화면에 나오는 부분은 우편의 머리부이다.

실행화면에 나오는 렬들은 편지상태, 편지번호, 보낸 사람, 날자와 시간, 행/문자수, 편지제목으로 되여있다.

편지상태

편지상태는 N/P/U/형태로 되여있는데 여기서 N은 새 통보문, P는 체계의 우편통에 저장된 우편물, U는 읽지 않은 우편물을 의미한다.

처음 수신된 편지부터 확인하려면 1을 쓰고 <Enter>를 누른다. 편지를 골라서 보려고 한다면 & 편지번호를 지정한다.

mail지령

mail의 여러 기능은 지령을 통하여 이루어진다. 지령입력방식은 "&"로 표시한다.

지 령	설 명
n	다음 편지로 이동한다.
f <편지번호>	편지목록을 보여준다. f * 또는 & f a 모든 편지목록을 볼수 있다.
d <편지번호>	지정한 편지번호의 편지를 지운다
s <편지번호>	파일에 지정한 편지내용을 보관한다.
u <편지번호>	편지를 지우지 않는다.
R <편지번호>	회답한다.
r <편지번호>	모두 회답한다.
q	mail을 완료한다.
x	mail을 완료한다. 체계우편통에서 편지들을 삭제하지 않는다.
h	통보문의 머리부부분을 출력한다.
!	쉘지령을 실행한다.
e	Ex방식으로 통보문을 편집한다.

회답보내기

편지보기지령창에서 현재우편에 회답하려고 한다면 R지령을 입력한다. 내용을 다 입력하면 [Ctrl+D]를 눌러 편지입력을 끝낸다.

주의 참조인이 없으면 그냥 <Enter>를 누른다.

편지보내기

mail을 실행한 상태에서 지령칸에 &m주소의 형식으로 입력하는것만으로도 편지를 송신할수 있다.

mail를 실행시키지 않은 상태에서 참조인(-c), 숨은 참조인(-b)을 포함한 우편을 송신해보자. 우편송신과정을 보려면 -v추가선택을 사용한다.

주의 편지를 쓰다가 내용을 수정하려고 할 때 새줄의 첫칸에 ~v를 입력하면 vi편집기상태에서 내용을 수정할수 있다.

mail지령으로는 2진파일을 직접 첨부할수 없다. 그러므로 파일을 부호화하여 본문형태로 전송할수 있게 만든다. 현재는 pine이나 netscape우편관리기, K우편프로그람 등 파일첨부가 자유로운 우편의뢰기를 많이 사용한다.

주의 본문안에 본문파일을 첨부할 때에는 새줄 첫번째 칸에 ~r 파일이름 형식으로 입력한다.

mailq /usr/bin/mailq

우편작업목록 보기

지령형식: mailq [추가선택]

-v : 대기렬의 내용을 자세히 보여준다.

sendmail을 통해 송신되는 우편은 대기렬에 쌓여서 송신대기상태에 들어간다. mailq 지령은 대기하고있는 우편목록을 보여준다.

지령실행결과 현시되는 우편목록의 매 마당의 내용은 다음과 같다.

Q-ID	우편의 대기렬ID를 보여준다.
Size	본문의 크기를 보여준다.
Priority	우편의 우선순위를 보여준다.
Q-Time	우편이 대기렬에 들어와 대기하기 시작한 시간을 보여준다.
Sender/Recipient	우편을 보내는 사람과 받는 사람을 보여준다. 또한 우편의 상태를 볼수 있다.

련관지령

sendmail-bp 발송될 우편의 대기상태를 보여준다.

man

/usr/bin/man

지도서보기

 지령형식: man [추가선택] [section] [-M path] [-P pager] [-S List] [-m system] [-p string] 지령

-a : 찾으려는 지령이 검색된 페지를 모두 보여준다.
-c : 사용자가 요구하는 지도서페지로 이동한다.
-d : 지도서를 보여주지 않고 지도서들이 있는 등록부들을 보여준다.
-D : -d추가선택과 같은 구성을 보여준 후 해당한 지도서페지를 보여준다.
-f : whatis지령과 같다. 즉 지령에 대한 간단한 설명을 보여준다.
-h : 도움말을 보여준다.
-k : apropos지령과 같다.
-K : 모든 지도서페지들에서 지정한 문자를 찾는다.
-t : /usr/bin/groff로 페지형식을 보여준다.
-w : 찾으려는 문자가 들어있는 페지의 경로를 보여준다.
-W : -w와 비슷하지만 추가정보가 없이 한 행에 하나씩 표시한다.
-C 파일이름 : 지도서페지의 설정파일을 지정한다. 기정파일은 /etc/man.config이다.
-M path : 지도서페지검색을 위한 경로를 지정한다.
-P 페지 : 지정한 페지를 지정한다.
-S 목록 : 두점으로 구분한 대화접속목록을 보여준다.
-m system : 주어진 체계이름에 의해서 지도서페지를 지정할수 있다.
-p string : nroff or troff의 앞에 실행하는 전처리기의 차례를 지정한다.

알고싶은 지령에 대한 사용지도서를 보여준다.

man지령은 Linux지령의 의미나 사용법을 모르는 경우에 사용하는데 man파일들을 검색하여 해당 지령에 대한 사용지도서를 보여준다.

만일 지정한 지령을 포함하고있는 지도서페지들을 모두 보려면 -K추가선택을 리용한다. 검색된 지령은 [ynq]중 하나의 대답을 선택하여 지도서보기(y), 지도서보지 않기(n), 탈퇴(q)를 결정할수 있다.

그러면 mtab와 관련된 파일을 -K추가선택으로 찾아보자.

지도서페지는 절들로 나누어져 있다. 지도서페지의 맨 웃부분에 있는 괄호안의 수자나 압축된 man파일이름의 끝수자가 바로 이 절을 나타낸다.

1	실행프로그람 혹은 쉘지령
2	체계호출 (핵심부제공함수)
3	서고호출 (체계서고포함함수)
4	특수파일 (대체로 /dev/의 파일)
5	파일류형모임 (례: /etc/passwd)
6	오락관련
7	마크로패키지집합
8	체계관리지령
9	핵심부부분프로그람

우의 실례에서 처음에 있는 ncpmount.8.gz파일은 끝수자가 8이므로 체계관리지령이라는것을 보여준다.

련관지령

apropos (man -k)
whatis (man -f)

mattrib

/usr/bin/mattrib

MS-DOS파일속성 변경

지령형식: mattrib [±추가선택] DOS파일 [DOS파일]

±a : 보존비트. 여벌복사프로그람이 새로운 파일을 지정할 때 사용된다.
±r : 읽기전용비트. 파일을 읽기전용으로 지정한다.
±s : 체계비트. MS-DOS가 체계파일을 가리킬 때 사용된다.
±h : 숨김파일비트. 숨긴 파일을 만들 때 사용된다.

MS-DOS파일의 속성을 변경한다.

속성의 변경은 요구하는 추가선택의 추가(+)와 제거(-)를 통해 이루어진다. 아래의 실례에서는 dosfile.txt파일에 읽기전용속성과 숨기기속성을 준다.

mdir지령의 모두 보기추가선택(-a)으로 숨긴 파일을 확인할수 있다.

mdel지령으로 숨긴 파일을 지우려면 Linux는 읽기전용파일을 삭제하겠는가를 사용자에게 문의한다.

mbadblocks

/usr/bin/mbadblocks

MS-DOS파일체계의 불량구역 표시

지령형식: mbadblocks 구동기:

MS-DOS파일체계로 형식화한 디스크의 불량구역을 검사한다.
플로피디스크를 MS-DOS파일체계인 fat로 형식화한 다음 불량구역을 검사할 때 리용한다.

```
[root@Dolphin root]# mbadblocks a:
plain_io: Input/output error
Bad cluster 18 found
plain_io: Input/output error
Bad cluster 19 found
plain_io: Input/output error
Bad cluster 20 found
```

18, 19, 20클라스터가 불량이라는것을 보여준다.

FAT파일체계 (MS-DOS파일체계)

파일체계는 조작체계가 파일을 보관하고 읽어들이는 규칙이다. 모든 조작체계에는 대체로 자기의 고유한 파일체계가 있다. 물론 필요에 따라 서로 다른 조작체계의 파일체계가 호환되는 경우도 있다.

Windows계렬에서는 FAT라는 파일체계를 사용한다. 원래는 FAT16을 사용하였지만 Windows 95 OSR-2 및 Windows 98에서는 성능이 훨씬 개선된 FAT32를 함께 사용하게 되였다.

mcd

/usr/bin/mcd

MS-DOS등록부 변경

지령형식: mcd [DOS등록부]

현재 사용하는 등록부를 변경한다.

mtools에 포함된 지령들은 지령파라메터로서 구동기를 DOS에서와 같이 A:, C:, D: 등으로 사용할수 있다. DOS등록부로 지정된 내용은 mcd지령으로 확인하거나 변경할수 있다.

```
[root@Dolphin root]# mcd
A:/
```

현재등록부가 플로피디스크등록부인 A:로 설정되여있다.

mtools지령중 하나인 mdir지령을 추가선택없이 사용하면 현재 mtools에서 지정되여 있는 등록부의 내용을 보여준다.

mcd로 DOS에서와 같이 작업등록부를 변경할수 있다.

mtools

mtools는 Linux나 UNIX에서 사용하는 MS-DOS지령이다. MS-DOS에서 사용하는 지령앞에 "m"을 붙여 만들었다. 사용법이 MS-DOS와 거의 비슷하므로 DOS를 사용한 사용자라면 쉽게 리용할수 있다. 뒤에 사용되는 추가선택도 거의 같다. 등록부 구분만은 "/"를 사용해야 한다.

/etc/mtools.conf파일에서 mtools의 지령에 사용될 설정을 변경할수 있다.

mtools.conf에는 BeOS, SCO Unix, Solarix, SunOS 등 여러 조작체계에서의 DOS등록부설정방법이 실례로 나와있다. 실례를 참고하면 쉽게 DOS등록부를 설정해서 사용할수 있다.

mcopy

/usr/bin/mcopy

Linux-DOS사이의 파일복사

 지령형식: mcopy [추가선택] [원본파일] [복사파일 혹은 등록부]

-t : 본문파일복사. DOS본문형식의 Line/Enter표시가 UNIX형식의 행표시로 변경된다. 또한 UNIX에서 DOS로 복사할 때는 그 반대로 변경된다.
-a : 본문(ASCII)파일복사. UNIX본문파일형식을 DOS형식으로 변경하거나 그 반대로 변경한다.
-T : 본문(ASCII)파일복사. 문자모임을 변경해준다.
-b : 묶음복사
-s : 반복복사
-p : 원본파일의 속성을 가진다.
-Q : 여러 파일을 복사하는 도중에 일정한 파일의 복사가 실패할 경우 복사를 중지한다.
-n : Linux나 UNIX파일을 덧쓰기할 때 경고통보문이 나오지 않는다
-m : 원본파일의 날자를 유지한다.
-v : 복사과정을 자세히 보여준다.

mcopy는 MS-DOS파일과 Linux/UNIX파일과의 복사지령이다. 물론 Linux/UNIX의 cp지령으로 복사는 가능하지만 mcopy는 DOS와 Linux/UNIX에서의 Line/Enter 형식을 변환하는 기능을 가지고있으므로 편리하게 리용할수 있다.

플로피디스크의 DosText.txt파일을 하드에 Linux/UNIX본문형식으로 보관하자.

```
# mcopy-t A:/DosText.txt /home/pirania
```

그냥 복사한 경우에는 Line/Enter형식이 서로 다르기때문에 잘못 인식하여 문장끝에 ^M 등의 기호가 찍히게 된다. 우와 같이 mcopy -t지령으로 DOS본문파일형식을 Linux/UNIX본문파일형식으로 변환할수 있다.

mdel　　　　/usr/bin/mdel

MS-DOS파일 삭제

지령형식: mdel [추가선택] DOS파일

-v : mtools의 판본정보를 보여준다.

MS-DOS파일체계에 있는 파일을 삭제하는데 리용한다.
다음의 실례는 MS-DOS형식으로 되여있는 플로피디스크에서 test.txt파일을 지운다.

등록부는 삭제할수 없으며 등록부를 정리하고싶을 때는 mdeltree 또는 mrd를 사용한다.

```
# mdel a:/test.txt
```

mdir지령으로 삭제된 파일을 확인할수 있다.

핵심부품파일완료를 알림

핵심부콤파일시간이 짧지 않으므로 많은 시간을 랑비하게 된다. 이 경우 조작탁의 《삑》하는 소리를 사용하여 작업완료를 알릴수 있다.

```
# make -s dep; make -s clean; make -s bzImage;\
make -s modules; make -s modules install; echo -e "\a"
```

소리가 잘 들리지 않는 환경에서는 XWindows를 리용하여 화면에 보여줄수 있다. 이때에는 다음과 같이 한다.

```
# make -s dep; make -s clean; make -s bzImage;\
make -s modules; make -s modules install; xmessage "complet!"
```

gnome환경이라면 gmessage를 사용할수도 있다.

mdeltree

/usr/bin/mdeltree

MS-DOS등록부 삭제

지령형식: mdeltree [-v] DOS 파일등록부

-v : 판본정보를 보여준다.

mdeltree는 mtools지령이다.
MS-DOS파일체계에 있는 등록부와 하위등록부의 모든 파일들을 삭제하는데 리용한다.
mdir지령으로 MS-DOS등록부의 파일목록을 확인한다.

a:/dosdir등록부전체를 한번에 지운다. mdeltree는 파일, 등록부를 지울수 있으며 하위등록부와 파일이 있는 경우에도 다 지워버린다.

mdel은 파일만 삭제할수 있다. 그러나 mdeltree는 등록부와 파일, 보조 등록부까지도 모두 삭제한다. 반면에 mrd는 등록부만 삭제하고 하위등록부나 파일은 삭제할수 없다.

mtools 삭제지령비교 용어상식

지령	삭제대상	설 명
mdel	파일	파일만 삭제할수 있다.
mdeltree	파일, 등록부	파일과 등록부를 지운다. 하위등록부나 파일이 있는 경우도 삭제가 가능하다.
mrd	등록부	등록부만 삭제할수 있다. 하위등록부나 파일이 있는 경우에는 삭제하지 못한다.

/usr/bin/mdir

MS-DOS파일목록 보기

지령형식: mdir [추가선택] DOS파일(등록부)

-/ : 파일목록을 보여준다.
-w : 파일이름만 보여준다.
-a : 숨긴 파일을 포함한 모든 파일을 보여준다.
-f : 디스크의 남은 용량을 검사하지 않고 파일목록을 보여준다. 따라서 속도가 빠르다.
-b : 파일이름과 등록부이름을 한 행에 하나씩 간단하게 보여준다.
-v : 판본정보를 보여준다.

mdir은 mtools지령이다.

MS-DOS파일체계에 있는 파일목록을 보여준다. 사용법과 추가선택은 MS-DOS와 거의 같다. 경로가 등록부가 아닌 경우에는 오유가 발생한다.

```
root@Dolphin:~
[root@Dolphin root]# mdir a:/

 Volume in drive A is SEC
 Volume Serial Number is 5E39-16F2
Directory for A:/

test                 383  02-17-2005  17:11  test
addtest              383  02-17-2005  17:11  addtest
addtest2             204  02-17-2005  17:11  addtest2
dosdir        <DIR>       02-17-2005  17:13  dosdir
        4 files                    970 bytes
                             1 455 104 bytes free
```

-a추가선택을 리용하면 숨긴 파일도 볼수 있다. -ab추가선택을 리용하여 숨긴 파일을 포함한 모든 파일을 한행씩 화면에 출력한다.

mformat

/usr/bin/mformat

MS-DOS파일체계 만들기

지령형식: mformat [-f 크기] [-1/4/8] 구동기
mformat [추가선택] 구동기

-f : 형식화할 DOS파일체계의 크기를 정한다.
-1 : single-sided에 형식화한다.
-4 : double-sided에 360KB형식화한다. "-1"과 같이 쓰면 한 방향에 180KB를 형식화한다.
-8 : 자리길당 8분구로 형식화한다.
-t sylinder : 실린더수를 정한다.
-h heading : 머리부수를 정한다.
-n sector : 자리길당 분구수를 정한다.
-v volume : 기록권표식을 해준다.

플로피디스크를 DOS파일형식으로 형식화한다.

mformat은 mtools지령이다. mformat는 UNIX파일체계에서 저수준으로 형식화된 디스크에 MS-DOS파일체계를 추가한다.

형식화할 플로피디스크의 용량과 특성

용량/KB	저장방향	자리길당 분구수	실린더	디스크 종류
160	single-sided	8	40	5.25
160	single-sided	9	40	5.25 DD
320	double-sided	8	40	5.25 DD
360	double-sided	9	40	5.25 DD
720	double-sided	9	80	3.5 DD
1200	double-sided	5	80	5.25 HD
1440	double-sided	8	80	3.5 HD
2880	double-sided	36	80	3.5 HD

플로피디스크를 1440KB의 DOS파일체계로 형식화하고 sec라는 이름을 부여해보자.

형식화된 디스크에 대한 자세한 정보를 보려면 minfo지령을 리용한다.

련관지령

mtools 참고

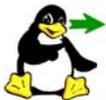
minfo

/usr/bin/minfo

MS-DOS파일체계에서 파라메터 보기

지령형식: minfo [추가선택] 구동기

-v : 기동분구를 16진수로 보여준다.

minfo는 mtools지령으로서 MS-DOS파일체계를 사용하는 디스크의 정보를 보여준다. 장치정보와 기동분구정보를 따로 보여준다.

련관지령

mtools　　　　참고

mkdir

/bin/mkdir

등록부만들기

지령형식: mkdir [추가선택] 등록부이름

-m, --mode : 새로 만들어지는 등록부의 권한을 설정한다.
-p, --parents : 상위경로를 같게 한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

mkdir은 등록부를 만들 때 사용하는 지령이다.

등록부를 만들면서 권한을 설정할수 있다. 설정하지 않으면 기정으로 755의 실행권한을 가진다.(chmod참고) 파일의 사용자와 그룹, 다른 사용자에게 똑같은 사용권한을 주

는 등록부를 만들어보자.

-p추가선택을 사용하여 상위등록부까지도 한번에 만들수 있다.

mke2fs

/sbin/mke2fs

ext2파일체계 생성

지령형식: mke2fs [추가선택] 장치이름

-b bytes : 블로크크기를 지정한다.
-c : 파일체계생성시 불량구역을 검사한다.
-f bytes : 쪼각(fragment)의 크기를 지정한다.
-i bytes : inode당 byte수를 정한다. 기정값은 4096byte이며 최소값은 1024byte이다.
-l filename : 파일에서 불량구역을 검사한다.
-m percent : 관리자에게 예약해둘 블로크의 백분률을 정한다. 기정값은 5%이다.
-q : 출력없이 실행한다. 스크립트안에서 사용되여 출력을 내보내지 않는다.
-v : 파일체계를 생성하는 과정을 자세히 보여준다.
-F : 지정한 장치가 블로크장치가 아니라도 mke2fs를 무조건 실행한다.
-s : 슈퍼블로크, 그룹서술자만을 쓴다.

하드디스크에 새로운 Linux파일체계를 만들 때 리용한다. mke2fs지령은 mkfs -t ext2 지령과 같다.

새로 생성된 파일체계는 탑재하여 사용할수 있다. /etc/fstab파일에 등록하면 체계를 재기동할 때마다 다시 탑재하지 않아도 된다.

련관지령

mkfs　　Linux파일체계생성지령

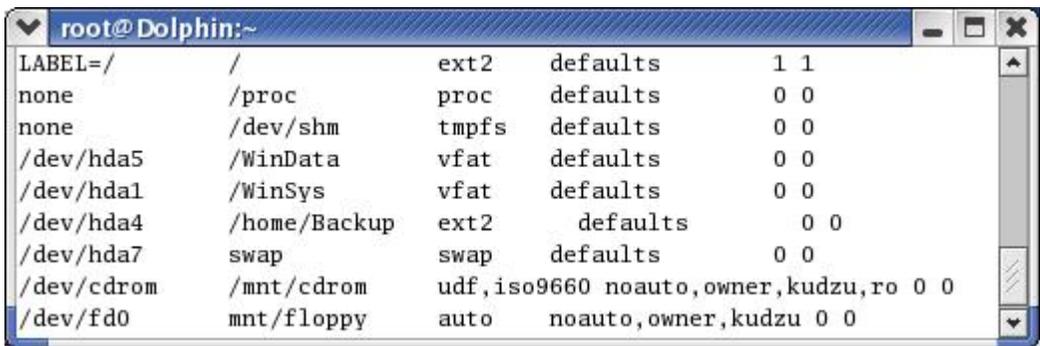
일반 파일체계

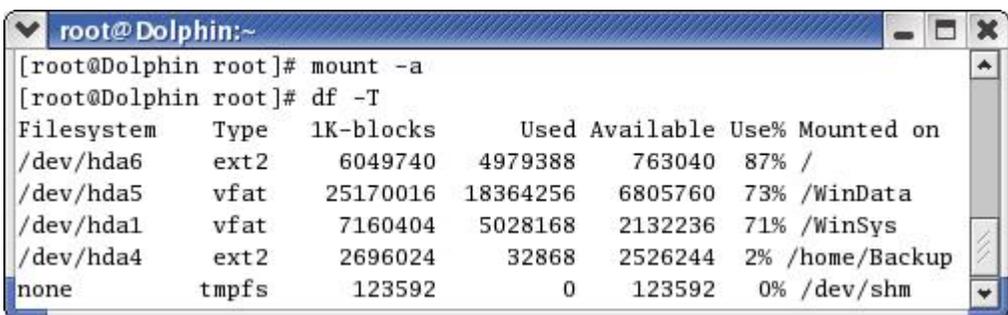
이전의 Linux체계에서는 파일체계로 ext2를 사용하였다. 체계가 비정상적으로 완료되였거나 기동될 때 파일체계를 검사한다. 그것은 파일이 작성되여 하드디스크에 기록된 다음 하드디스크의 색인정보를 갱신하기전에 체계가 비정상적으로 꺼지면 이 파일에 대한 정보를 알수 없기때문이다. 하드디스크의 용량이 커짐에 따라 이러한 작업도 상당히 오랜 시간이 걸리게 되였다.

일반 파일체계는 파일을 작성하고 하드디스크의 색인정보가 갱신되지 않아도 자체로 기록파일과 색인정보를 비교하여 오유를 수정하므로 안전하면서도 용량이 큰 하드디스크를 사용할 때 매우 편리하다. 현재 Linux에서 사용할수 있는 일반 파일체계로는 Reiserfs, JFS(IBM), XFS(SGI), ext3 등이 있다

fstab파일

fstab파일에는 체계기동시 자동적으로 탑재되는 파일체계가 설정되여있다. 그러므로 기동할 때마다 하드디스크를 반복추가하지 않고 자동적으로 탑재하여 사용하자면 /etc/fstab파일을 수정하여야 한다.

/etc/fstab파일을 편집기로 연다. mke2fs 등의 지령으로 파일체계가 새로 만들어진 하드디스크를 체계기동때마다 자동적으로 읽게 하려면 아래와 같이 새로 추가해야 할 파일체계의 정보를 적어준다. 이때 탑재하여 사용할 등록부인 /home/Backup은 만들어져있어야 한다.

```
root@Dolphin:~
LABEL=/          /              ext2     defaults        1 1
none             /proc          proc     defaults        0 0
none             /dev/shm       tmpfs    defaults        0 0
/dev/hda5        /WinData       vfat     defaults        0 0
/dev/hda1        /WinSys        vfat     defaults        0 0
/dev/hda4        /home/Backup   ext2       defaults        0 0
/dev/hda7        swap           swap     defaults        0 0
/dev/cdrom       /mnt/cdrom     udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0         mnt/floppy     auto     noauto,owner,kudzu 0 0
```

체계가 기동할 때 저장한 fstab정보를 자동적으로 읽어 탑재하며 체계를 완료하지 않고 사용하려면 mount -a지령을 사용한다. df -T지령으로 현재 체계에 탑재되여있는 파일체계정보를 확인한다.

```
root@Dolphin:~
[root@Dolphin root]# mount -a
[root@Dolphin root]# df -T
Filesystem    Type   1K-blocks      Used Available Use% Mounted on
/dev/hda6     ext2     6049740   4979388    763040  87% /
/dev/hda5     vfat    25170016  18364256   6805760  73% /WinData
/dev/hda1     vfat     7160404   5028168   2132236  71% /WinSys
/dev/hda4     ext2     2696024     32868   2526244   2% /home/Backup
none         tmpfs      123592         0    123592   0% /dev/shm
```

mkfs

/sbin/mkfs, /sbin/mkfs.msdos, /sbin/mkfs.reiserfs, /sbin/mkfs.bfs, /sbin/mkfs.ext2/sbin/mkfs.vfat

Linux파일체계 생성

지령형식: mkfs [추가선택] 장치이름 [크기]

-V : 정보를 자세히 보여준다.
-t type : 만들어질 파일체계의 종류를 지정한다.
-c : 파일체계를 생성하기전에 먼저 디스크의 불량구역을 검출한다.
-l filename : 지정한 파일이름으로부터 디스크의 불량구역목록을 읽는다.
-v : 작업상태와 결과를 자세히 보여준다.

mkfs는 ext2, ext3, vfat 등의 여러가지 파일체계를 만들수 있다.
Primary Slave구획의 구성상태를 살펴보자. fdisk -l지령으로 확인할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# fdisk -l /dev/hda

Disk /dev/hda: 40.0 GB, 40020664320 bytes
240 heads, 63 sectors/track, 5169 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1       949   7174408+   b  Win95 FAT32
/dev/hda2           950      5169  31903200    f  Win95 Ext'd (LBA)
/dev/hda5           950      4280  25182328+  83  Linux
/dev/hda6          4281      5093   6146248+  83  Linux
/dev/hda7          5094      5169    574528+  82  Linux swap
```

실례로 /dev/hda4구획을 ext2파일체계로 생성하려면 다음과 같이 새로 파일체계를 만들 장치이름을 입력한다.

```
# mkfs -t ext2 /dev/hda4
```

련관지령

mke2fs　　ext2파일체계생성지령
mkdosfs　　MS-DOS파일체계생성지령
mkswap　　Linux교환령역을 지정하는 지령

mknod

/bin/mknod

특수파일만들기

지령형식: mknod [추가선택] 파일이름 {bcu} Major번호 Minor번호
　　　　　mknod [추가선택] 파일이름

-m, --mode : 방식은 사용자권한을 말한다. chmod지령을 참고하여 방식을 사용한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

특수파일을 만든다. /dev/sda파일을 살펴보자.

brw-rw----에서 b는 블로크장치를, c는 문자장치를 나타낸다. 여기서 블로크장치는 디스크, CD-ROM과 같이 매체를 리용하여 블로크단위의 검색을 진행하는 장치이며 문

자장치는 통신포구, 인쇄기포구 등 한 byte씩 순차적호출을 하는 장치를 말한다.

8, 0은 Major번호와 Minor번호를 나타낸다. SCSI디스크는 Major번호가 8로 시작하며 IDE디스크라면 3으로 시작한다. sda1, sda2는 Major번호는 8파 갈고 Minor번호가 1, 2로 증가한다.

이 /dev/의 하드웨어장치파일은 mknod지령이나 /dev/MAKEDEV스크립트를 리용한다.

mkswap

/sbin/mkswap

교환령역지정

지령형식: mkswap [추가선택] 장치이름 [블로크크기]

-c 교환령역을 만들기 전에 먼저 하드의 불량구역을 검사한다.

mkswap지령은 지정한 장치나 파일을 교환령역으로 지정한다. 아래의 실례는 512MB크기의 Linux교환파일을 만드는것이다.

512MB의 교환파일 만들기

free지령으로 작성된 교환파일을 확인한다.

```
# free
```

교환령역을 해제하려면 swapoff지령을 사용한다.

```
# swapoff /swap
# free
```

mlabel

/usr/bin/mlable

MS-DOS디스크표식 만들기

지령형식: mlabel [추가선택] 구동기:

-c : 현재 설정된 표식을 없앤다.
-s : 현재 설정된 표식을 보여준다.
-n : 새로운 계렬번호를 우연수로 설정한다.
-N serial : 새로운 계렬번호를 지정하여 만든다. 16진수 8자리로 만든다.

이 지령은 mtools지령이다. MS-DOS의 label지령과 같은 기능으로 쓰인다. 플로피디스크의 표식을 새롭게 지정하려면 구동기이름 "a:"를 입력한다.

기록권의 계렬번호는 16진수 8자리로 지정한다.

```
# mlabel -N 01234567 a:
```

변경된 내용은 mdir지령을 리용하여 확인할수 있다.

련관지령

mtools 참고

DOS (Disk Operating System)

DOS는 PC들에서 이미전에 널리 사용되였던 조작체계이다. DOS는 Microsoft 회사가 개발한 지령행방식의 대면부를 가지는 조작체계이므로 사용하기가 불편하였으며 지령들을 일일이 기억해야 하였다.

그러나 Windows의 초기판본은 실제로는 MS-DOS상에서 실행되는 일종의 편의 프로그람과 같은 형태로 배포되였으며 지금도 Windows에서는 DOS prompt화면을 자주 리용한다.

mmd

/usr/bin/mmd

MS-DOS등록부만들기

지령형식: mmd [추가선택] 등록부이름

MS-DOS등록부를 만든다.

mmd는 mtools지령으로서 MS-DOS등록부를 만든다.

련관지령

mtools 참고

mmount

/usr/bin/mmount

MS-DOS파일체계 탑재

 지령형식: mmount DOS구동기 탑재위치

mmount는 mtools지령이다.

mtools가 사용하고있는 플로피디스크구동기는 A구동기를 /mnt/floppy등록부에 탑재하여 사용한다.

```
# mmount a: /mnt/floppy
```

mmount로 MS-DOS구동기를 등록하기 위해서는 /etc/fstab 또는 /etc/mtab파일에 등록되여있어야 한다.

만약 등록되여있지 않으면 탑재위치의 등록부를 파일내용에서 찾을수 없다는 오유통보문을 출력한다. 이때에는 아래의 내용을 /etc/fstab파일에 추가한다.

```
/dev/fd0    /mnt/floppy    vfat    defaults    0    0
```

련관지령

mtools 참고

mmove

/usr/bin/mmove

MS-DOS파일 옮기기/이름바꾸기

 지령형식: mmove [추가선택] 파일 [이동위치/파일이름]

-V : 판본정보를 보여준다.

mmove는 mtools지령이다.

이 지령은 파일이나 등록부를 이동하거나 이름을 바꾼다. mmove지령은 mtools를 통해 지정된 구동기에만 파일이나 등록부를 이동할수 있다. 구동기를 지정하지 않으면 mtools

가 지정하는 등록부를 의미한다. 등록부는 mcd지령을 리용하여 변경한다.

련관지령

mtools 참고

modprobe

/sbin/modprobe

핵심부모듈적재

지령형식:

모듈적재

modprobe [-a -n -v] [-C config] [-t 형태] 패턴 OR 모듈이름1 모듈이름2

모듈목록보기

modprobe [-l] [-C config] [-t 형태] 패턴

설정보기

modprobe [-C config] -c

모듈제거 혹은 자동제거

modprobe [-C config] -r [모듈이름 ...]

-a, --all : 모듈과 의존성으로 련관된 다른 모듈도 같이 적재한다.
-c, --showconfig : 현재의 설정을 본다.
-d, --debug : 오유수정방식.
-h, --help : 도움말을 보여준다.
-k, --autoclean : autoclean모듈을 적재한다.

-l, --list : 핵심부모듈들을 보여준다.
-n, --show : 실지 동작을 수행하지 않는다.
-q, --quiet : 결과를 보여주지 않는다.
-r, --remove : 지정된 모듈을 제거한다.
-s --syslog : 통보문을 syslog파일에 보관한다.
-t, --type moduletype : 지정한 형태의 모듈들만 보여준다.
-V, --version : 판본정보를 보여준다.
-C, --config configfile : /etc/modules.conf파일대신 지정한 설정파일을 리용한다.

modprobe지령은 insmod보다 높은 수준의 조종을 진행할수 있는 모듈조종지령이다. 모듈등록부는 /lib/modules/[핵심부판본]/kernel/에 있다.

현재기억기에 적재된 모듈들을 먼저 보도록 하자.

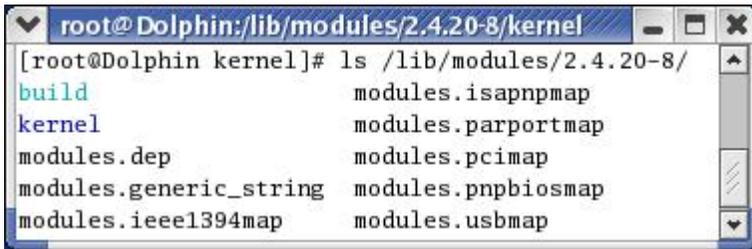
```
root@Dolphin:/lib/modules/2.4.20-8/kernel
[root@Dolphin kernel]# lsmod
Module              Size  Used by    Not tainted
ide-cd             35708   0  (autoclean)
cdrom              33728   0  (autoclean) [ide-cd]
i830               74336   1
agpgart            47776  11  (autoclean)
parport_pc         19076   1  (autoclean)
lp                  8996   0  (autoclean)
parport            37056   1  (autoclean) [parport_pc lp]
autofs             13268   0  (autoclean) (unused)
e100               60644   1
microcode           4668   0  (autoclean)
nls_iso8859-1       3516   1  (autoclean)
nls_cp437           5116   1  (autoclean)
vfat               13004   1  (autoclean)
fat                38808   0  (autoclean) [vfat]
keybdev             2944   0  (unused)
mousedev            5492   1
hid                22148   0  (unused)
input               5856   0  [keybdev mousedev hid]
usb-uhci           26348   0  (unused)
ehci-hcd           19976   0  (unused)
usbcore            78784   1  [hid usb-uhci ehci-hcd]
ext3               70784   1
jbd                51892   1  [ext3]
```

그중에서 cdrom모듈을 기억기에서 제거해보자.

```
# rmmod cdrom
```

insmod지령으로 다시 cdrom모듈을 적재한다.

```
# insmod /lib/modules/2.4.20-8/kernel/net/ipv4/netfiter/cdrom.o
```

depmod지령으로 매 모듈에 대한 의존성을 검사하여 모듈자료기지를 생성한다. 생성된 파일은 /lib/modules/[핵심부판본]/modules.dep로 보관된다.

실례로 아래의 ipt_MASQUERADE.o모듈을 적재하자면 먼저 ip_conntrack.o, ip_tables.o, iptable_nat.o모듈을 적재해야 하는데 modprobe지령은 의존성을 자동적으로 검사하여 같이 적재해준다.

아래에 modules.dep파일에서 ipt_MASQUERADE.o와 관련된 모듈들을 보여준다.

```
/lib/modules/2.4.20-8/kernel/net/ipv4/netfilter/ipt_MASQUERADE.o:
/lib/modules/2.4.20-8/kernel/net/ipv4/netfilter/ip_conntrack.o \
/lib/modules/2.4.20-8/kernel/net/ipv4/netfilter/ip_tables.o \
/lib/modules/2.4.20-8/kernel/net/ipv4/netfilter/iptable_nat.o
```

more

/bin/more

화면단위로 파일보기

 지령형식: more [추가선택] [-행수] [+/표현식] [+행수] [파일이름]

-d : 파일을 한 화면에 표시한 후에 "Press space to continue, 'q' to quit"내용을 보여주고 해당 한 건을 누르지 못하면 "Press 'h' for instructions"이라는 내용을 보여준다.
-p : 화면의 내용을 모두 지우고 본문을 보여준다.
-s : 빈 행이 여러개이면 한개 행만 보여준다.

-u : 밑줄속성을 가진 문자를 그대로 보여준다.
+/표현식 : 정규표현식이 발견되는 부분부터 보여준다.
-행수 : 한 화면에 출력할 행수를 지정한다.
+행수 : 지정한 행부터 보여준다.

more지령은 한 페지이상인 내용을 페지별로 볼수 있게 해준다. 또한 표현식에 의한 검색을 진행할수 있으며 요구하는 페지로 이동할수 있다.

more상태에서 사용할수 있는 지령

h	more의 도움말을 출력하며 계속해서 현시하려면 공백건을 누른다.
z	다음 페지를 출력한다.
[Enter]건	한개 행씩 출력한다.
d, [Ctrl+D], [F2]	반페지씩 출력한다.
q, Q	more를 완료한다.
f	다음 페지를 출력한다.
b, [Ctrl+B]	이전 페지를 출력한다.
!<cmd>, :!<cmd>	지정하는 지령을 실행할수 있다.
/검색할 단어	지정한 단어가 있는 내용을 현시한다.

ps지령을 실행하면 현시하는 내용이 한페지를 넘게 된다. 그러므로 흐름선을 리용하여 more지령으로 한페지씩 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# ps aux | more
USER       PID %CPU %MEM   VSZ  RSS TTY   STAT START   TIME COMMAND
root         1  0.0  0.0  1368   72 ?     S    11:53   0:04 init [5]
root         2  0.0  0.0     0    0 ?     SW   11:53   0:00 [keventd]
root         3  0.0  0.0     0    0 ?     SW   11:53   0:00 [kapmd]
root         4  0.0  0.0     0    0 ?     SWN  11:53   0:00 [ksoftirqd_CPU0]
root         9  0.0  0.0     0    0 ?     SW   11:53   0:00 [bdflush]
root         5  0.0  0.0     0    0 ?     SW   11:53   0:00 [kswapd]
root         6  0.0  0.0     0    0 ?     SW   11:53   0:00 [kscand/DMA]
root         7  0.0  0.0     0    0 ?     SW   11:53   0:01 [kscand/Normal]
root         8  0.0  0.0     0    0 ?     SW   11:53   0:00 [kscand/HighMem]
root        10  0.0  0.0     0    0 ?     SW   11:53   0:00 [kupdated]
root        11  0.0  0.0     0    0 ?     SW   11:53   0:00 [mdrecoveryd]
root        15  0.0  0.0     0    0 ?     SW   11:53   0:00 [kjournald]
--More--
```

mount

/bin/mount

장치련결

지령형식: mount -a [추가선택] 장치 등록부

-h : 도움말을 보여준다.
-V : 판본정보를 보여준다.
-a : fstab파일에 등록되여있는 모든 파일체계를 다 탑재한다. noauto추가선택이 설정된 파일체계는 제외된다.
-v : 자세한 설명을 보여준다.
-f : 실제 체계호출은 하지 않고 탑재할수 있는가를 점검한다.
-n : /etc/mtab파일에 쓰기를 하지 않고 탑재한다. /etc가 읽기전용 파일체계인 경우에 필요하다.
-r : 읽기만 가능하게 탑재한다. -o ro추가선택과 같다
-w : 읽기/쓰기방식으로 탑재한다. 기정설정값이다. -o rw추가선택과 같다
-t type : 파일체계의 형식을 지정한다.
-o : 이 추가선택뒤에 ":"과 함께 구체적인 선택을 위한 다음의 추가선택들을 리용할수 있다.
. async : 파일체계에 대한 I/O가 비동기적으로 진행되도록 한다.
. auto : -a추가선택으로 탑재한다.
. defaults : rw, suid, dev, exec auto, nouser, async를 기본추가선택으로 한다.
. dev : 파일체계상의 문자장치나 블로크특수장치를 해석한다.
. exec : 2진파일을 실행할수 있다.
. noauto : -a추가선택으로는 탑재되지 않는다.
. nodev : 파일체계상의 문자, 블로크장치에 대한 해석을 하지 않는다
. noexec : 파일체계에 실행권한을 주지 않는다.
. nosuid : set-UID, set-GID를 무시하게 한다.
. nouser : 일반사용자는 탑재를 못하게 한다.
. remount : 이미 탑재된 파일체계를 다시 탑재한다.
. ro : 파일체계를 읽기만 가능하게 한다.
. rw : 읽기/쓰기방식으로 탑재한다.
. suid : set-UID, set-GID가 효력을 낼수 있게 해준다.
· vsync : 파일체계에 대한 I/O가 동기적으로 진행되도록 한다.
· user : 일반사용자도 탑재할수 있게 한다.

Linux에서는 모든 장치까지도 /밑에 하나의 등록부로 인식한다. 그리고 그 장치들을 /etc/fstab에 등록하여 기동시 자동탑재되게 한다.

우와 같이 매 구획을 지정하여 기동시에 자동으로 탑재를 하게 하였다. /etc/fstab에는 [장치이름] [탑재할 등록부] [파일체계] [추가선택] [덤프] [기동시 파일체계의 점검순서]로 지정되여있다.

자세한 설명은 다음과 같다.

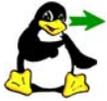
장치이름	탑재될 블로크장치 또는 원격파일체계
탑재할 등록부	파일체계를 탑재할 탑재위치인 등록부를 지정한다.
파일체계	파일체계의 류형을 지정한다.
추가선택	파일체계와 관련된 탑재추가선택을 지정한다.
덤프	dump지령에 의하여 덤프될 파일체계를 지정한다. 만일 0이면 덤프가 필요없다고 판단한다.
기동시 파일체계의 점검순서	기동시 파일체계에 이상이 있는가를 확인한 후 이상이 있을 경우에는 체계검사를 순서대로 진행한다. 0은 점검하지 않는다는 의미이다. 보통 1은 /(root)구획에 대하여 지정하며 나머지는 2이상의 우선순위를 지정하면 된다.

-l추가선택은 탑재한 파일체계의 정보를 보여준다.

/dev/hdc1파일체계를 탑재해제해보자.

```
# umount /hdc1
```

다시 /dev/hdc1파일체계를 탑재해보자.

```
# mount /dev/hdc1 /hdc1
```

련관지령

/etc/fstab	탑재할 파일체계와 추가선택을 담고있는 파일
/etc/mtab	현재 탑재된 파일체계와 추가선택을 담고있는 파일
umount	탑재해제지령

mrd

/usr/bin/mrd

MS-DOS등록부삭제

지령형식 : mrd [추가선택] MS-DOS등록부

-V : 판본정보를 보여준다.

mrd는 MS-DOS의 등록부를 삭제할 때 리용하는 mtools지령이다.

삭제하려는 등록부안에 파일이 있거나 하위등록부가 있을 때에는 삭제되지 않는다. 또한 이 지령으로는 파일을 삭제하지 못한다. 이때에는 mdeltree지령을 사용하여 하위등록부나 파일이 포함되여있는 등록부를 삭제할수 있다.

mren

/usr/bin/mren

MS-DOS파일의 이름바꾸기

지령형식 : mren [추가선택] 파일이름 [변경할 파일이름]

-a : 중복되는 이름과 비슷한 이름으로 자동변경한다.
-A : 중복되는 이름과 비슷한 이름으로 자동변경한다. (모든 파일 적용)
-r : 중복되지 않는 이름으로 바꾼다.
-R : 중복되지 않는 이름으로 바꾼다. (모든 파일 적용)
-o : 덧쓰기한다.
-O : 덧쓰기한다. (모든 파일 적용)
-s : 지령을 적용시키지 않는다.
-S : 지령을 적용시키지 않는다. (모든 파일 적용)
-V : 판본정보를 보여준다.

이 지령은 MS-DOS파일이나 등록부의 이름을 변경하는데 리용하는 mtools지령이다.

subdir라는 등록부의 이름을 testdir로 변경해보자.

```
# mren a:/subdir a:/testdir
```

다음의 실례는 test파일이름을 test1로 변경한다. 파일이름이 같을 경우 자동적으로 다른 이름으로 변경되도록 하려면 -A추가선택을 준다.

```
# mren -A a:/test a:/test1
```

mdir지령으로 변경된 파일과 등록부이름을 확인할수 있다.

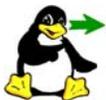
```
root@Dolphin:~
[root@Dolphin root]# mdir a:/
 Volume in drive A has no label
 Volume Serial Number is 4D60-BE24
Directory for A:/

test1-1            383 02-17-2005  18:17  test1-1
testdir      <DIR>     02-17-2005  18:13  testdir
test1              383 02-17-2005  18:17  test1
        3 files                 766 bytes
                          1 455 616 bytes free
```

련관지령

mtools 참고

mtype

/usr/bin/mtype

MS-DOS파일의 내용보기

 지령형식 : mtype [추가선택] 파일이름

-t : 본문파일을 보여준다. CR/LF형식을 LF형식으로 변경하여 출력한다.
-v : 판본정보를 보여준다.

mtype는 MS-DOS파일을 보여주는 mtools지령이다.

```
# mtype -t a:/test.txt
```

우와 같이 지령을 주면 해당한 파일의 내용이 출력된다. -t는 MS-DOS의 행시작형식을 Linux/UNIX의 행시작형식으로 변경하여 출력한다.

carriage return / line feed (CR/LF)

한행의 출력을 끝내고 다음행의 첫 부분으로 보내는 기능. carriage return은 복귀기능으로서 현재행의 맨 앞으로 보내며 line feed는 다음행의 같은 위치로 보낸다. 두 기능이 합쳐지면 행의 마지막까지 출력하고 다음행의 맨 앞부분으로 이동하게 된다.

mv

파일이동

 지령형식 : mv [추가선택] source dest
mv [추가선택] source directory
mv -d [추가선택] --target-directory=directory sourceenv
[추가선택] [변수=값...] [지령 [인수...]]

-b : 여벌복사파일을 생성한다.
-i, --interactive : .bashrc에 지적되여있는 추가선택으로서 이동할 때 같은 파일이 있으면 사용자에게 확인을 요구한다.
-v, --verbose : 이동하기 전에 파일이름들을 출력한다.
-u, --update : 이동할 파일이 새로운 파일인 경우에만 이동한다.

mv지령은 파일 또는 등록부를 이동하거나 이름을 바꿀 때 리용한다. 원천과 대상의 이름이 다르면 이름을 변경하는것으로 된다. 이동할 파일이 여러개인 경우에는 이동방식으로만 동작한다.

아래의 실례는 이름변경의 기능을 수행하는 mv지령이다.

```
root@Dolphin:/WinData/Song/music
[root@Dolphin music]# ls
2.mp3            come.mp3     tp5362.mp3    track0~8.mp3  track1~7.mp3
Audio CD_01.MP3  friend.mp3   track0~2.mp3  track0~9.mp3  track1~8.mp3
Audio CD_02.MP3  love.mp3     track0~3.mp3  track1~2.mp3
Audio CD_03.MP3  oil.mp3      track0~5.mp3  track1~4.mp3
Audio CD_04.MP3  oi.mp3       track0~6.mp3  track1~5.mp3
Audio CD_05.MP3  red.mp3      track0~7.mp3  track1~6.mp3
[root@Dolphin music]# mv Audio CD_01.MP3 1.mp3
```

현재등록부에서 .mp3으로 끝나는 파일을 모두 /root등록부로 이동시켜보자.

```
[root@Dolphin music]# mv *.mp3 /root/
[root@Dolphin music]# ls /root/*.mp3
/root/1.mp3        /root/tp5362.mp3    /root/track1~2.mp3
/root/2.mp3        /root/track0~2.mp3  /root/track1~4.mp3
/root/come.mp3     /root/track0~3.mp3  /root/track1~5.mp3
/root/friend.mp3   /root/track0~5.mp3  /root/track1~6.mp3
/root/love.mp3     /root/track0~6.mp3  /root/track1~7.mp3
/root/oil.mp3      /root/track0~7.mp3  /root/track1~8.mp3
/root/oi.mp3       /root/track0~8.mp3
/root/red.mp3      /root/track0~9.mp3
```

namei

/usr/bin/namei

련결원천파일의 경로보기

지령형식 : namei [추가선택] 경로이름 [경로이름...]

-x : 탑재된 파일체계의 경로에 대해서는 "d" 대신 "D"를 표시한다.
-m : rwxrx-rx의 형태로 호출권한을 보여준다.

지정한 파일의 형태와 구체적인 정보를 보여주는 지령이다. 이 지령은 파일이나 등록부의 이름을 절대경로로 지정하면 뿌리등록부로부터 하위등록부까지의 경로를 하나하나 보여주고 지정된 파일의 형태가 기호련결파일인가, 블로크형장치인가 혹은 일반파일인가 하는 정보를 보여준다.

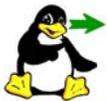
출력되는 행의 첫 부분에 이 파일의 형태를 알리는 문자를 표시하는데 그것은 다음과 같다.

- f: 입력된 경로이름
- d 경로의 개별적인 등록부들
- l 기호련결 (련결대상이 된 파일은 다음에 들여쓰기하여 보여준다.)
- s 소케트
- b 블로크형장치
- c 문자형장치
- \- 일반파일
- 알수 없는 파일

httpd.conf파일을 한번 살펴보자. 입력된 경로이름이 f:다음에 표시되고 d가 붙었으므로 /, etc, httpd, conf는 이 파일의 경로에 있는 등록부들이라는것을 알수 있으며 httpd.conf에는 -가 붙었으므로 그것이 일반파일이라는것을 알수 있다.

```
root@Dolphin:~
[root@Dolphin root]# namei /etc/httpd/conf/httpd.conf
f: /etc/httpd/conf/httpd.conf
 d /
 d etc
 d httpd
 d conf
 - httpd.conf
```

-m추가선택을 리용하면 해당한 등록부나 파일의 호출권한까지도 함께 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# namei -m /etc/httpd/conf/httpd.conf
f: /etc/httpd/conf/httpd.conf
 drwxr-xr-x /
 drwxr-xr-x etc
 drwxr-xr-x httpd
 drwxr-xr-x conf
 -rw-r--r-- httpd.conf
```

련관지령

ls　　등록부의 내용을 보여주는 지령

stat　　파일이나 파일체계의 상태를 보여주는 지령

netstat

/bin/netstat

망의 상태보기

지령형식 : netstat [추가선택]

-a, --all : 모든 소케트의 상태를 보여준다.
-c, --continuous : 1s에 한번씩 정보를 련속적으로 보여준다.
-i, --interface : 체계의 망대면부장치정보를 추가적으로 보여준다.
-n, --numeric : 국부주소와 목적지주소를 령역형태가 아닌 IP주소형태로 된 망주소로 보여준다.
-o, --timers : 망통신시계와 관련된 정보를 추가적으로 보여준다.
-p, --program : 매 소케트에 포함된 프로그람의 PID와 이름을 보여준다.
-r, --route : 핵심부경로배정표를 보여준다.
-s, --statistics : 매 UDP, TCP, ICMP, IP규약의 주요통계를 보여준다.
-t, --tcp : TCP소케트만 보여준다.
-u, --udp : UDP소케트만 보여준다.
-V, --version : 판본정보를 보여준다.
-h, --help : 도움말을 보여준다.

netstat지령으로는 체계에 련결된 망과 관련되는 정보를 자세하게 볼수 있다.

실례를 통하여 구체적으로 보기로 하자.

netstat -nr추가선택으로 경로배정표를 볼수 있다. 이것은 route지령과 같은 기능을 수행한다.

```
root@Dolphin:~
[root@Dolphin root]# netstat -nr
Kernel IP routing table
Destination    Gateway        Genmask          Flags  MSS Window  irtt Iface
192.168.8.0    0.0.0.0        255.255.255.0    U        0 0          0 eth0
169.254.0.0    0.0.0.0        255.255.0.0      U        0 0          0 eth0
127.0.0.0      0.0.0.0        255.0.0.0        U        0 0          0 lo
0.0.0.0        192.168.8.83   0.0.0.0          UG       0 0          0 eth0
```

아래에 매 마당의 내용을 보여주었다.

마당이름	설명
Gateway	경로배정항목에서의 관문
Genmask	경로배정항목의 망마스크
Flags	경로배정에 관한 여러가지 기발을 표시
Flags U	대면부가 우에 놓인 상태
H	경로배정을 통한 콤퓨터에로의 련결
G	관문의 경로배정
D	경로배정기가 재지정되여 동적으로 생성
M	경로배정이 ICMP지정통보문을 통해 수정되였을 경우
MSS	최대토막크기(Maximum Segment Size)
Window	원격으로 한번에 보내는 자료의 수신량
irtt	initial round trip time으로서 0은 초기설정값을 사용한다는 의미
Iface	망대면부

netstat -i추가선택은 매 망대면부장치의 망상태 및 통계를 보여준다.

```
[root@Dolphin root]# netstat -i
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0    1683   0      0      0     1413   0      0      0 BMRU
lo    16436   0   62116   0      0      0    62116   0      0      0 LRU
```

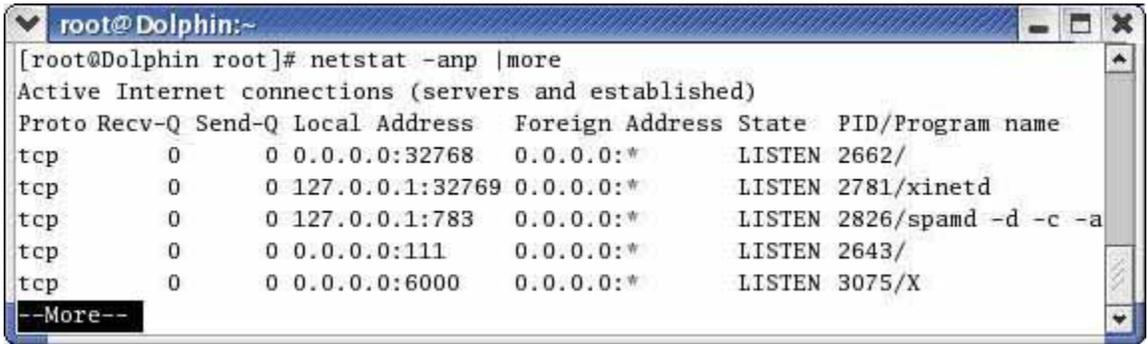
다시 말하여 eth0, lo 등 매 대면부의 최대전송단위(MTU-Maximum Transfer Unit)값과 송수신파케트의 오유나 넘침상태 등을 살펴볼수 있다.

마당의 의미는 다음과 같다.

마당이름	설명
Iface	망대면부
MTU	최대전송단위 (Maximum Transmission Unit)
Met	거리값(메터)
RX-OK/RX-ERR	파케트수신/파케트수신오유
RX-DRP/RX-OVR	파케트수신손실/파케트수신 넘침상태
TX-OK/TX-ERR	파케트송신/파케트송신오유
TX-DRP/TX-OVR	파케트송신손실/파케트송신 넘침상태

netstat -ta추가선택은 TCP소케트와 관련된 모든 봉사기의 목록을 보여준다.

```
# netstat -ta
```

netstat -anp추가선택은 흔히 망련결상태를 표시하는데 많이 리용한다.

Program name에 대한 매 State가 LISTEN이면 이 체계에서 봉사되고있는 망봉사로 볼수 있다. 여기에서는 xinetd, spamd 등 데몬의 State상태가 LISTEN으로서 봉사상태이며 매 봉사포구는 Send-Q마당의 값이 모두 0이다.

newgrp /usr/bin/newgrp

그룹변경

 지령형식 : newgrp [그룹]

newgrp지령은 현재 사용자가 속해있는 그룹을 바꾸는 지령이다.

새 파일을 사용자의 그룹이 아닌 다른 그룹의 파일로 만들자면 파일을 생성한 다음에 chown지령으로 그룹을 지정하여야 하지만 newgrp을 사용하면 변경된 그룹속성으로 파일이 생성된다.

newgrp에서 지정하는 그룹은 실지로 존재하는 그룹이름이나 GID여야 한다. 현재 존재하는 그룹이름이나 GID는 /etc/group파일에서 찾아볼수 있다.

touch지령으로 빈 파일을 생성하는 경우에 파일의 그룹은 소유자의 기본그룹대로 정해진다. newgrouptest1이라는 이름을 가진 파일을 touch지령으로 생성해보자.

```
# touch newgrouptest1
```

그러면 newgrouptest1파일은 소유자와 그룹이 다 root로 설정되게 된다.

newgrp지령으로 현재사용자그룹을 admin으로 변경한 다음 다시 파일을 생성해보자.

```
# newgrp admin
# touch newgrouptest2
```

```
root@Dolphin:~
[root@Dolphin root]# ls -al newgrouptest2
-rw-r--r--    1 root     admin           0 Feb 17 18:41 newgrouptest2
```

현재사용자의 그룹이 admin으로 변경되였으므로 새로 만든 파일의 그룹도 admin로 된다.

UNIX

UNIX는 1969년에 벨연구소에서 개발한 시분할체계이다. UNIX라는 이름은 그 이전체계인 MULTIX에 기반을 두고 지은것으로 알려져있다. 1974년에 UNIX는 C 언어로 다시 만들어졌으며 여러 제공업체와 대학 그리고 프로그람개발자들에 의해 많은 확장판과 추가사항이 생겨나 다양한 판본의 UNIX들이 출하되였다. UNIX는 특정한 콤퓨터회사가 독점하는 조작체계가 아니였으며 표준프로그람작성언어로 작성되고 대중적인 창안들을 받아들여 부단히 개선됨으로써 최초의 공개형표준조작체계로 발전하였다.

newusers

/usr/bin/newusers

사용자추가

지령형식 : newusers 파일

newusers는 체계지령으로서 /etc/passwd파일로부터 사용자를 추가하거나 사용자 정보를 갱신한다. 하지만 pwconv지령으로 /etc/shadow파일이 생성되였다면 이 지령은 수행되지 않는다. 현재 사용중인 거의 모든 체계는 보안을 위해 shadow파일을 사용하므로 newusers는 거의 리용되지 않는다.

shadow

adduser지령으로 사용자계정을 추가하면 /etc/passwd파일에 사용자이름과 ID, 암호화된 비밀통과어, 쉘과 사용자home등록부 등의 정보가 보관된다. 체계를 여러 사용자가 리용하는 경우 모든 사용자가 /etc/passwd파일을 읽을수 있어야 등록가입을 할수 있으므로 passwd파일은 모두에게 공개된것이나 다름없다. 이러한 문제를 해결하기 위해 만들어진것이 pwconv(shadow)이다. pwconv는 암호화된 비밀통과어만 /etc/shadow파일에 보관하고 허가권을 제한함으로써 아무 사용자나 통과어를 볼수 없도록 하였다.

nice

/bin/nice

우선순위변경

지령형식 : nice [추가선택][지령 [인수]…]

-n num, -num, —adjustment=num : 지령의 우선권에 10이 아니라 지정한 수값을 더한다.
--help : 사용법을 보여준다.
--version : 판본정보를 보여준다.

일정작성우선권을 변경하여 프로그람이 동작할 때 할당되는 순위를 변경하는 지령이다. 만약 아무런 인수도 주지 않으면 상속받은 현재의 일정작성우선권을 출력한다. 인수가 있다면 조정된 일정작성을 먼저 실행한다. 조정수값이 생략되면 지령의 우선권은 10으로 설정된다. nice에 의해 조정될수 있는 범위는 -20(가장 높은 우선권)에서부터 19(가장 낮은 우선권)까지이다. 체계관리자만이 조정수값을 부수로 정할수 있다.

nice지령은 프로그람의 실행시 우선순위를 변경하지만 renice는 실행된 프로쎄스가 가지게 될 우선순위를 지정한다. nice는 정수값에는 "-"를 부수값에는 "--"를 사용하지만 renice는 정수값에 "-"를 사용하지 않는다.

vi를 배경프로쎄스로 두개정도 실행해보자.

```
# vi &
# vi &
```

다음 ps aux지령으로 vi의 프로쎄스ID를 알아보자.

이제는 renice지령으로 이 매 프로그람의 우선권을 변경해본다.

renice -3지령으로 4007프로쎄스의 우선순위를 -3으로 한다.

renice -2지령으로 4008프로쎄스에는 -2의 우선순위를 부여한다.

4008프로쎄스의 우선순위를 다시 6으로 변경할수도 있다.

ps l 4007 4008지령으로 우선순위를 확인해보자. 각각 -3과 6의 우선순위를 가지고 있다.

```
[root@Dolphin root]# ps l 4007 4008
F   UID   PID  PPID PRI  NI   VSZ  RSS WCHAN  STAT TTY        TIME COMMAND
0     0  4007  3971  17  -3  7832 2304 finish T<   pts/0      0:00 vim
0     0  4008  3971  26   6  7828 2316 finish TN   pts/0      0:00 vim
```

nice지령으로 실행시 우선순위를 지정할수도 있다.

nice 7지령으로 7의 우선순위를 부여해보자.

```
# nice 7 vi &
```

그럼 방금 실행한 vi의 PID와 우선순위를 확인해본다.

```
root@Dolphin:~
[root@Dolphin root]# ps aux | grep vi
root      4007  0.0  0.9  7832 2304 pts/0    T<   18:41   0:00 vim
root      4008  0.0  0.9  7828 2316 pts/0    TN   18:41   0:00 vim
root      4055  0.0  0.3  3940  796 pts/0    TN   18:49   0:00 vi
root      4057  0.0  0.2  3568  624 pts/0    S    18:49   0:00 grep vi
```

nl

/usr/bin/nl

행번호달기

지령형식 : nl [추가선택] [파일]

-b, --body-numbering=STYLE : 행번호형식을 지정한다. 사용가능한 STYLE은 아래에서 설명한다.
-d, --selection-delimiter=CC : 지정한 CC를 페지의 구분으로 한다. 기정은 \:이다.
-f, --footer-numbering=STYLE : 페지바닥부행에 지정한 STYLE을 적용한다. 기정 STYLE은 n이다.
-h, --header-numbering=STYLE : 페지머리부행에 지정한 STYLE을 선택한다. 기정 STYLE은 n이다.
-i, --page-increment=num : 지정한 수값만큼 페지행번호를 증가시킨다.
-n, --number-format=FORMAT : 행번호의 서식을 지정한 FORMAT로 한다. 사용가능한 FORMAT는 아래에 있다.
-p, --no-renumber : 페지가르기를 진행한 다음 번호달기를 새로 시작하지 않는다.
-s, --number-separator=string : 지정한 문자렬로 본문과 행번호를 분리한다.
-v, --first-page=num : 지정한 수값으로 첫번째 페지행번호를 설정한다.
-w, --number-width=num : 지정한 수값으로 한행의 문자수를 지정한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

nl지령은 지정한 파일을 읽어 파일의 행번호를 자동으로 달고 표준출력으로 보여주는 지령이다. 이와 같은 기능을 가진 지령으로는 cat -b가 있다. 아래에 지정할수 있는 STYLE 류형들을 보여준다.

- a : 모든 행에 행번호를 출력한다.
- t : 빈 행을 제외한 나머지 행들에 번호를 출력한다.
- n : 행번호를 출력하지 않는다.
- pREGEXP : 정규표현식에 맞는 행에만 번호를 출력한다.

다음은 지정할수 있는 FORMAT류형들을 보여준다.

- ln : 행번호의 빈자리에 0을 채우지 않고 왼쪽맞추기하여 보여준다.
- rn : 행번호의 빈자리에 0을 채우지 않고 오른쪽맞추기하여 보여준다.
- rz : 행번호의 빈자리에 0을 채우면서 오른쪽맞추기하여 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# nl /etc/sensors.conf | more
     1  # Sensors configuration file used by 'libsensors'
     2  #-----------------------------------------------
     3  #
     4  ################################################################
     5  #                                                              #
     6  # PLEASE READ THIS HELPFUL HINT!!!                             #
     7  #                                                              #
     8  #     The 'set' lines (generally for min and max values)       #
     9  #     do not take effect until you run 'sensors -s' as root !!! #
    10  #     We suggest you put 'sensors -s' in a /etc/rc.d/... file   #
    11  #     to be run at boot time after the modules are inserted !!! #
    12  #                                                              #
    13  ################################################################
    14  #
    15  #
    16  # OVERVIEW
    17  # --------
--More--
```

주의 cat -n으로도 같은 결과를 얻을수 있다.

nohup

/usr/bin/nohup

등록탈퇴시 프로그람실행

지령형식: nohup 지령 [인수...]

등록탈퇴를 하는 동안에도 실행중인 지령을 정지하지 않고 수행하게 하는 지령이다. 보통 현재 진행하는 작업이 많은 시간을 요구할 때 효과적으로 사용할수 있다.

nohup는 Hangup신호를 무시하면서 지령의 우선권을 5만큼 증가시켜 가입탈퇴후에도 계속 배경방식으로 실행할수 있게 한다.

등록탈퇴한 후 배경방식에서 지령을 수행하려면 다음과 같이 입력한다.

```
$ nohup find / -name *.c & > result &
```

뿌리등록부에서 find지령을 실행시킬 때 검색시간이 오래 걸릴수 있으므로 nohup지

령으로 가입탈퇴후에도 계속 실행할수 있다.

주의 xlock와 같은 화면보호프로그람은 화면잠금기능(Screen Lock)도 동시에 제공하고있다. 그러나 다른 사람이 XWindows에서 동작하는 어떤 프로그람을 리용하여 자기의 체계에 접근할수 있으므로 오랜 시간 자리를 비워야 할 경우에는 nohup지령을 리용하는것이 좋다.

nslookup

/usr/bin/nslookup

이름봉사 질문

지령형식: nslookup [추가선택...] [host_to_find | -봉사기]

령역이름봉사기에 질문하여 결과값을 돌려주는 지령이다.

/etc/resolv.conf에 지정되여있는 이름봉사기를 기정값으로 인식하여 질문을 보낸다. 이름봉사기에 대한 자세한 설명은 데몬설명부분을 참고하기 바란다.

비대화방식으로 찾으려는 주콤퓨터의 주소를 질문할수 있다. 만일 아래와 같은 오유통보문이 나온다면 이름봉사설정이 제대로 안된 결과이므로 관련설정을 살펴보아야 한다.

아래와 같이 이름은 linuxroot.co.kp이고 주소는 192.168.8.83으로서 우에서 본 /etc/resolv.conf파일의 내용과 같으므로 이름봉사기의 설정에 이상이 없음을 알수 있다.

대화방식으로 DNS레코드류형에 대한 자세한 질문을 할수 있다. nslookup지령을 실행한 다음에 나타나는 ">"입력부분에 이름봉사기에 대한 질문을 set type형태로 지정하여 자세한 정보를 볼수 있다.

먼저 set type=SOA방식으로 linuxroot.co.kp에 질문해보자. 이것을 통해 이름봉사기의 Foreword zone파일의 SOA레코드값들을 살펴볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# nslookup linuxroot.co.kp
Note: nslookup is deprecated and may be removed from future releases.
Consider using the dig' or host' programs instead. Run nslookup with
the -sil[ent]' option to prevent this message from appearing.
> set type=type
unknown query type: type
> linuxroot.co.kp
Server:     192.168.8.83
Address:    192.168.8.83 #53

Non-autyoritative answer:
Name:   linuxroot.co.kp
Address:    192.168.8.83
> set type=SOA
> linuxroot.co.kp
Server:     192.168.8.83
Address:    192.168.8.83 #53

linuxroot.co.kp
        origin=ns.linuxroot.co.kp
        mail addr=root.ns.linuxroot.co.kp
        serial=2002031200
        refresh=28800
        retry=14400
        expire=3600000
        minimum=86400
>
```

set type=MX방식으로 MX레코드를 질문해보자. 이름봉사설정의 우편봉사기는 mail.linuxroot.co.kp로 되여있는것을 확인할수 있다.

type ANY추가선택으로 주어진 주콤퓨터이름과 관련된 모든 자원레코드를 볼수 있다.

set type에 지정할수 있는 type값은 다음과 같다.

A	주콤퓨터의 주소
ANY	대부분 관련된 정보
CNAME	별명과 관련된 CNAME(Canonical name) 레코드정보
HINFO	주콤퓨터의 CPU와 조작체계를 지정하는 HINFO 레코드정보
MD	MD(Mail destination) 레코드정보
MG	MG(Mail group member) 레코드정보

MINFO	우편통이나 우편목록과 관련이 있는 MINFO(Mailbox or mailst information) 레코드정보
MR	MR(Mail rename) 레코드정보
MX	MX(Mail exchanger) 레코드정보
NS	NS(Nameserver) 레코드정보
PTR	주콤퓨터이름지정 PTR(program trouble report)레코드정보
SOA	SOA(Domain start-of-authority) 레코드정보
UINFO	UINFO(User Information) 레코드정보

od

/usr/bin/od

8진수로 파일보기

지령형식: od [추가선택] [파일]...

-a : 1byte를 문자로 표시한다.
-b : 1byte를 부호없는 8진수로 표시한다.
-c : 1byte를 ASCII문자 혹은 탈퇴문자를 사용하여 표시한다.
-d : 2byte를 부호없는 10진수로 표시한다.
-f : 4byte를 류동소수점으로 표시한다.
-h : 2byte를 16진수로 표시한다.
-i : 2byte를 10진수로 표시한다.
-l : 4byte를 10진수로 표시한다.
-o : 2byte를 8진수로 표시한다.
-x : 2byte를 16진수로 표시한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일을 요구하는 형식으로 보여준다.
실례로 작은 본문을 2byte씩 16진수와 10진수로 출력해보자.

찍어내기(dump)

주기억장치의 내용을 출력한다. 기억장치의 내용을 확인하여 오유수정하는데 많이 쓰인다.

진법과 수의 표현

binary - 2진수

2진수는 "0"과 "1"이라는 오직 2가지 종류의 수자로만 구성된 수체계로서 콤퓨터에서 자료표현에 리용된다.

사람들은 0부터 9까지 모두 10개의 수자로 구성된 10진법을 써왔으며 그보다 큰 수자들은 10개의 수자를 조합하여 만든다. 그러나 콤퓨터는 오직 0과 1로만 구성된 2진법체계만을 인식하고있다. 하지만 10진법에서 할수 있는 모든 연산(더하기, 덜기, 곱하기, 나누기)들이 2진법에서도 가능하다.

10진법에서는 매 수자의 위치가 10의 제곱형태로 표현된다. 실례로 456은 다음과 같이 나타낼수 있다.

100 이 4 개+10 이 5 개+1 이 6 개, 즉 4×100+5×10+6×1

이와 비슷한 리치로 2진법에서도 매 수자의 위치는 2의 제곱으로 표현된다. 실례로 2진수 1101은 다음과 같이 표현할수 있다.

8 이 1 개+4 가 1 개+2 가 0 개+1 이 1 개, 즉 1×8+1×4+0×2+1×1

octal- 8진법

Octal은 8진법을 가리키는 용어이다. 8진법에서는 수들이 0, 1, 2, 3, 4, 5, 6 및 7 등 8개의 수자를 리용하여 구성된다. 7다음의 수는 10으로, 17다음의 수는 20 등으로 바뀐다. 프로그람작성에서 8진수는 2진수에 비해 짧게 표기할수 있다는 장점으로 하여 2진수대신에 사용되는 경우가 있다.

hexadecimal- 16진수

16진수란 16을 기준으로 하는 수체계를 말한다. 16진법에서는 수자 0~9까지와 영어문자 A~F까지를 사용한다. 아래의 표에 모두 같은 값을 가지는 2진수와 10진수 그리고 16진수를 표시하였다.

16진수는 한 byte가 항상 8bit로 정의되는 근대콤퓨터에서 2진수를 표현하는 편리한 방법이다. 콤퓨터기억기의 내용을 보여줄 때 (새로운 프로그람을 오유수정하기 위하여 기억기dump를 하거나 또는 본문문자렬을 표현할 때, 또는 2진수값의 문자렬을 프로그람이나 HTML페지에 써넣을 때 등) 한 자리의 16진수는 4자리의 2진수를 표시할수 있다. 두개의 16진수는 8bit 즉 한 byte를 표현할수 있다.

passwd

/usr/bin/passwd

통과어변경

지령형식: passwd [추가선택] [사용자]

-l : 지정한 사용자에게 열쇠를 걸어 더 이상 접근하지 못하게 한다.
-u : -l추가선택으로 열쇠가 걸린 사용자에게서 열쇠를 풀어준다.
-d : 사용자의 통과어를 삭제한다.
-S : 지정한 사용자의 통과어를 암호화한 정보를 보여준다.

passwd지령은 사용자의 통과어를 변경한다.

체계관리에서 통과어는 보안과 관련하여 매우 중요하다. passwd지령을 주기적으로 사용하여 통과어를 변경함으로써 콤퓨터를 보다 더 보안할수 있다. 또한 체계관리자(root)는 자신뿐만아니라 일반사용자의 통과어도 임의로 변경할수 있다.

한가지 주의할것은 useradd지령으로 사용자를 추가한 후 반드시 passwd지령으로 통과어를 생성해야 한다는것이다. 그렇지 않으면 추가된 사용자는 등록가입할수 없게 된다. 아래에서는 root로 등록가입하였으므로 관리자의 통과어를 변경하게 된다.

```
[root@Dolphin root]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

관리자는 사용자의 통과어를 변경할수도 있다.

passwd admin지령으로 admin사용자의 통과어를 변경하여보자.

```
[root@Dolphin root]# passwd admin
Changing password for user admin.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

passwd지령과 밀접한 련관이 있는 /etc/passwd파일을 한번 보기로 하자.

/etc/passwd파일은 username:password:uid:gid:gecos:homedic.shell의 형태로 되여있다.

```
[root@Dolphin root]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
..........................
webmaster:x:500:500::/home/webmaster:/bin/bash
admin:x:501:501::/home/admin:/bin/bash
webmin:x:502:502::/home/webmin:/bin/bash
```

매 마당은 다음과 같은 의미를 가진다.

username	고유한 계정을 구별하는 사용자식별자

password	사용자의 통과어가 암호화된 형태로 나타나며 사용자가 읽을수 있다. 보안상 위험이 있으므로 pwconv지령으로 /etc/shadow파일을 사용한다.
uid	사용자 ID로서 계정을 구별하는 값이다.
gid	그룹 ID로서 /etc/group과 관련된다.
gecos	사용자의 실제이름, 주소, 전화번호 등의 정보가 있다.
homedir	사용자의 home등록부
shell	사용자가 등록가입할 때의 쉘

paste

/usr/bin/paste

파일합치기

지령형식: paste [추가선택] 파일...

-d, —delimiters=char : 지정한 char를 렬지시자로 설정한다.
-s, —serial : 한개 파일을 한개 행으로 만들어 합친다.
—help : 도움말을 보여준다.
—version : 판본정보를 보여준다.

paste지령은 지정된 파일들의 해당한 행을 합쳐 표준출력으로 보여주는 지령이다. 실례를 통하여 한번 보기로 하자. 먼저 test1파일과 test2파일을 각각 살펴보자.

paste지령은 이 두개의 파일들을 하나로 만들어 표준출력으로 보여준다.

-s추가선택을 사용하면 매 파일에 있는 내용이 한개 행으로 출력된다.

아래의 실례에서는 ls지령으로 /에 있는 모든 등록부를 한 행에 출력하되 paste -d 추가선택으로 ";"를 매 렬에 대한 구분자로 표시하였다.

GNOME(GNU Network Object Model Environment)

GNOME은 UNIX기반의 조작체계사용자를 위한 도형사용자대면부와 콤퓨터탁상응용프로그람들의 모임이다. 사용자에게 편리하고 익숙한 외형과 함께 여러 응용프로그람들을 함께 제공하여 사용자가 UNIX/Linux를 쉽게 사용할수 있도록 한다.

또한 GNOME은 프로그람작성자들이 GNOME사용자대면부를 사용하는 응용프로그람들을 개발하는데 리용할수 있는 서고를 포함하고있다.

pathchk

/usr/bin/pathchk

파일이름검사

지령형식: pathchk [추가선택] 파일이름

-p, --portability : 현재파일체계가 아닌 모든 POSIX체계들에 대한 검사를 제공한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

pathchk지령은 파일이름이 유효하며 다른 체계에로의 이식성이 있는가를 검사하는 지령으로서 다음 조건에 만족하면 "0"을 내보내며 그렇지 않으면 "1"을 내보낸다.

- 파일의 완전경로에 속한 모든 등록부가 검색(실행)허가권을 가지고있지 않는 경우.
- 파일이름의 길이가 파일체계가 지원하는 최대파일길이보다 큰 경우.
- 존재하는 등록부이름에 대응하는 매 파일이름의 길이가 그 파일체계에서 지정한 최대길이보다 긴 경우.

-p추가선택을 리용하면 검색되는 파일이 POSIX규격에 맞는가 확인해볼수 있다. 아래의 실례와 같이 파일이름이 18문자인것은 POSIX규격인 14문자를 넘기때문에 다른 체계에 이식하는데서 문제가 발생할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# pathchk -p /root/install.log.syslog
pathchk: name `install.log.syslog' has length 18; exceeds limit of 14
```

본문파일행사이에 빈 행 추가

본문파일에서 행사이의 간격이 없으면 문서를 읽기가 힘들다. 이것을 극복하기 위하여 다음과 같은 방법으로 행 사이에 빈 행을 넣어줄수 있다.

이렇게 하면 in.txt파일의 매 행사이에 빈 행이 추가된 out.txt파일이 만들어진다.

```
# awk '{printf(%s\n\n"),$0}' in.txt > out.txt
```

프로그람실행시간 보기

프로그람이 실행되는 정확한 시간을 알고 싶을 때에는 다음과 같이 한다. 아래는 핵심부콤파일과정의 한 실례이다.

```
# time -v sh -c 'make zlilo'
```

이식성

현재체계에서 동작하고있는 프로그람이 다른 체계에서도 오유없이 동작한다면 이것을 이식성이 좋다고 말한다.

pidof /sbin/pidof

PID확인

지령형식: pidof [추가선택] 프로그람

-s : 지정한 프로그람의 PID가 여러개 있을 경우에는 하나만 보여준다.
-x : 스크립트와 함께 실행되는 프로그람이면 그 쉘의 id도 함께 보여준다.
-o pids : 찾으려고 하는 프로그람이름중 pids로 지정된 PID만 제외하고 나머지를 보여준다.

pidof지령은 실행중인 프로그람의 PID를 찾는다. 이것은 프로세스를 만든 지령은 알지만 해당 PID를 모르는 경우에 매우 유용하게 사용할수 있다.

ps aux지령과 grep지령을 결합하여 vi와 관련된 프로세스만을 추출해보자. 그러면 vi에 대한 자세한 프로세스정보들을 살펴볼수 있다.

하지만 vi프로쎄스에 대한 PID만 필요한 경우 pidof지령이 편리하다.

이에 대한 보다 구체적인 설명은 nice지령을 참고하기 바란다.

ping

/bin/ping

망련결확인

지령형식: ping [추가선택] host

-b : 동시통보주소로 ping지령을 보낸다.
-c count : 지정한 수만큼 파케트를 보낸후 끝낸다.
-f : flood ping파케트를 보낼수 있다.
-i wait : 지정한 간격으로 파케트를 보낸다.
-I interface_address : 발신지의 대면부주소를 지정한다.
-l preload : ping을 보내기전에 지정한 수만큼 파케트를 먼저 보낸다.
-n : ping을 보내는 콤퓨터이름을 IP주소형태로 보여준다.
-p pattern : 송신할 파케트를 채우기 위한 16 "pad" byte를 지정한다. 이것은 망상에서 자료와 관련된 문제를 진단하는데 리용된다. 례를 들어 -p ff는 전송파케트를 모두 1로 채운다.
-q : ping파케트의 반응상태를 보여주지 않고 통계만 보여준다.
-r : 경로배정표를 리용하지 않고 직접 련결된 콤퓨터에 파케트를 보낸다.
-s packetsize : 지정한 크기만한 파케트를 보낸다. 기정값은 56이며 최대값은 65507이다.
-v : ECHO_RESPONSE와 함께 수신된 ICMP파케트의 자세한 정보를 보여준다.
-V : 판본정보를 보여준다.

ICMP ECHO_REQUEST파케트를 콤퓨터에 전송하여 망련결성을 확인하는 지령이다. 망에 련결된 하나의 콤퓨터가 다른 콤퓨터에 ICMP파케트를 보내게 되면 파케트는 목적콤퓨터에 도착한 시간을 기록한 후 다시 파케트를 보낸 콤퓨터로 돌아오게 된다. 이 두 콤퓨터사이에 오고 간 파케트의 왕복시간을 계산하여 콤퓨터망의 련결성이나 콤퓨터의 동작상태를 알수 있다. 만일 응답이 없다면 망련결상태나 대상콤퓨터에 이상이 발생한것으로 판단할수 있다.

-c추가선택으로 전송할 파케트수를 지정한다. 다음 실례에서는 파케트를 세번 보내는 추가선택을 사용하였다. 파케트의 왕복시간이 검사되는것으로 192.168.8.84콤퓨터와의 련결상태가 안전하다는것을 확인할수 있다.

다음은 192.168.8.85에 파케트를 보내는 실례이다. 응답이 없으므로 망상에 문제가 있는것으로 예측할수 있다.

주의 보안상의 리유로 ICMP파케트의 응답을 막아놓는 경우도 있다.

ping에 대한 응답을 하지 않는 방법

아래와 같이 지령을 입력하면 콤퓨터에 ping반응이 되지 않아 ping지령에 의한 확인이 불가능하다.

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

주콤퓨터가 있는 broadcast ping에만 반응을 하지 않을수도 있다.

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

이 지령을 재기동한 후에도 계속 적용하려면 /etc/rc.d/rc.local파일이나 /etc/sysctl.conf파일에 다음의 내용을 추가하여 넣는다.

```
# cat /etc/sysctl.conf
x
net.ipv4.icmp_echo_ignore_broadcasts = 1
netipv4.icmp_echo_ignore_all = 1
```

ICMP

인터네트조종통보문규약(Internet Control Message Protocol)으로서 Internet에서 오유통보문의 생성 및 검사 그리고 IP와 관련된 정보를 제공하는 통신규약을 말한다. 실례로 ping지령은 Internet접속검사를 위하여 ICMP를 사용한다.

파케트 (packet)

망에서 자료전송을 위한 하나의 전송자료단위이다.

파케트의 크기는 수신측과 송신측에 의하여 사용되는 통신규약에 의하여 결정된다. 일반적인 표준은 128byte이다.

pr

/usr/bin/pr

파일인쇄설정

지령형식: pr [파일]

-num_cols, --columns=num=num_cols : 지정한 수값만큼 렬수를 설정한다.(기정값은 1)
-a, --across : --columns와 함께 사용하여 렬들을 아래로가 아니라 수평으로 출력한다.
-c, --show-control-chars : 인쇄시 보이지 않는 문자들을 인쇄가능한 문자로 변환한다.
-d, --double-space : 공백을 중복공백으로 출력한다.

-D, --date-format=FORMAT : 머리부의 날자를 지정한 FORMAT로 한다.
-e [char[width]], --expand-tabs[=char[width]] : Tab를 공백문자로 변환한다. 만일 char를 지정하였다면 지정한 문자를 Tab문자로 한다.(기정은 Tab문자) 만일 너비값을 지정한다면 지정한 값을 Tab문자크기로 설정한다.(기정값은 8)
-F, -f, --form-feed : 새로운 페지에 출력하기 위한 행시작문자대신 용지공급문자(form feed)를 사용한다.
-h header, --header=header : 머리부문자렬을 지정한다.
-i [char[width]], --output-tabs[=char[width]] : 공백을 Tab문자로 바꾼다. 만일 문자를 지정하였다면 지정한 문자를 Tab문자로 한다.(기정은 Tab문자) 만일 너비값을 지정한다면 지정한 값을 Tab문자크기로 설정한다.(기정값은 8)
-J, --join-lines : 모든 렬을 합친다. -W추가선택을 무시한다.
-l page_length, --length=page_length : 페지길이를 지정한 값으로 설정한다. 기정값은 66행이다.
-m, --merge : 모든 파일들을 평행으로 출력한다.
-n [delimiter[digits], --number-lines[=delimiter[digits] : 행번호를 단다. 만일 구분기호가 지정된다면 구분자를 지정된 값으로 설정한다.(기정값은 Tab) 만일 너비값이 지정된다면 지정된 자리수를 기준으로 행번호를 단다.(기정값은 5)
-o width, --indent=width : 지정한 값만큼 매 행을 들여쓰기한다.(기정값은 0)
-r, --no-file—warning : 입력된 파일을 열수 없더라도 경고통보문을 표시하지 않는다
-s [delimiter], —separator[=delimiter] : 구분기호로 지정된 한개 문자로 렬을 구분한다. (기정값은 Tab)
-t, --omit-header : 페지머리부와 페지바닥부를 생략한다.
-T, --show-nonprinting : -t추가선택과 비슷하나 용지공급문자까지 제거한다.
-v, --show-nonprinting : 인쇄되지 않는 문자를 인쇄가능한 문자로 변환한다.
-w page_width, --width=pagewidth : 지정한 값으로 페지행수를 설정한다.(기정값은 72)
-W page_width, --page-width=page_width : 지정한 값으로 페지행설정을 바꾼다.(기정값은 72)
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

본문파일을 인쇄가능한 형태로 페지가르기를 하여 표준출력으로 내보내는 지령이다. 여기에는 페지번호, 날자, 시간 및 파일이름을 가진 머리부가 포함된다.

pr지령을 리용하여 /etc/sendsors.conf파일을 인쇄가능한 형태로 변경해보자. 머리부부분에서 인쇄날자와 본문파일이름, 페지번호를 볼수 있다.

"pr 파일이름"지령은 화면에 내용을 보여주므로 lpr지령을 사용하여 직접 인쇄할수도 있다.

```
# pr test.txt | lpr
```

-h추가선택은 머리부를 지정한 문자렬을 인쇄한다.

```
# pr -h "TEST PAGES" test.txt | lpr
```

여러개의 파일을 련속 인쇄할수도 있다.

```
# pr -m -h "TEST PAGES" test.txt text2.txt | lpr
```

포스트스크립트(Postscript)

포스트스크립트는 인쇄된 페지의 모양을 지정하는 언어이다. 일반적으로 인쇄기 제작자들은 주요 조작체계들에서 실행될수 있는 포스트스크립트 쏘프트웨어가 내장되여있거나 또는 적재할수 있도록 인쇄기를 만든다.

포스트스크립트파일은 확장자가 .ps로 끝나므로 쉽게 구분할수 있다. 또한 포스트스크립트파일은 Adobe PDF형식으로 쉽게 변환하여 사용할수 있다.

praliases

/usr/sbin/praliases

우편별명보기

지령형식: praliases [추가선택]

-C file : 지정한 파일에서 우편설정파일을 읽는다. 기정파일은 /etc/sendmail.cf이다.
-f file : 지정한 file에서 우편별명파일을 읽는다. 기정파일은 /etc/aliases이다.

특정한 계정에 도착하는 우편을 다른 사람에게 넘겨주는 우편별명설정내용을 확인하는 지령이다.

가장 널리 사용하는 우편봉사기는 Sendmail봉사기이다. 이 봉사기는 Internet전자우편의 표준규약인 SMTP(Simple Mail Transfer Protocol)를 통하여 우편봉사를 진행한다. 우편별명은 우편사용자의 별명이라고 생각하면 된다. 만일 한 우편사용자가 admin계정과 webmaster계정, webmin계정 등 여러가지 계정들을 관리한다면 매 우편사용자를 생성하는것보다 그것들을 하나의 계정으로 별명화하는것이 편리하다. 별명설정내용을 볼수 있는것이 praliases지령이다.

아래에서 보는바와 같이 /etc/aliases파일은 우편별명을 위한 설정파일이다. 여기에서는 bin, daemon, adm 등의 계정으로 들어오는 우편물을 모두 root계정에서 받아볼수 있게 하였다.

printenv

/usr/bin/printenv

환경변수보기

지령형식: printenv [환경변수이름]

printenv지령은 환경변수들의 값을 보여준다. 이름을 지정하면 지정한 환경변수의 내용만을 보여주며 지정하지 않으면 모든 환경변수의 값을 보여준다.

```
[root@Dolphin root]# printenv
SSH_AGENT_PID=4359
HOSTNAME=Dolphin.co.kp
PVM_RSH=/usr/bin/rsh
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
GTK_RC_FILES=/etc/gtk/gtkrc:/root/.gtkrc-1.2-gnome
WINDOWID=18874627
QTDIR=/usr/lib/qt-3.1
USER=root
```

export지령으로도 아무런 추가선택을 지정하지 않거나 -p추가선택을 주면 쉘에서 반출된 모든 환경변수의 목록을 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# export
declare -x BASH_ENV="/root/.bashrc"
declare -x COLORTERM="gnome-terminal"
declare -x DISPLAY=":0.0"
declare -x GDMSESSION="Default"
declare -x GNOME_DESKTOP_SESSION_ID="Default"
declare -x GTK_RC_FILES="/etc/gtk/gtkrc:/root/.gtk
rc-1.2-gnome2"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTSIZE="1000"
declare -x HOME="/root"
```

ps

/bin/ps

프로세스상태보기

지령형식: ps [추가선택]

전체적인 프로세스와 관련된 추가선택

-A : 모든 프로세스를 보여준다.
-N : -A추가선택과 비슷하나 이미 실행한 ps프로세스만은 제외한다.
-a : 대화접속지시기 및 말단기와 관련되지 않은 프로세스를 제외한 모든 프로세스를 보여준다.
-d : 대화접속지시기를 제외한 모든 프로세스를 보여준다.
-e : 핵심부프로세스를 제외한 모든 프로세스를 보여준다.
T : 현재말단의 모든 프로세스를 보여준다.
a : 현재말단의 사용자 고유프로세스를 보여준다.
r : 현재 실행중인 프로세스를 보여준다.
x : 말단이 없는 프로세스를 보여준다.
--deselect : -N추가선택과 같다.

특정프로세스를 선택하여 지정한 목록만 보여주는 추가선택

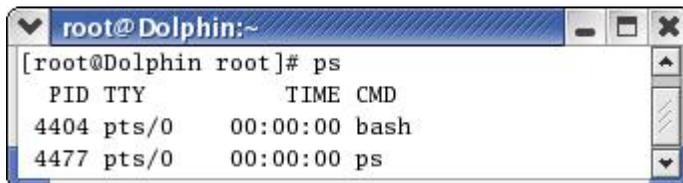
-C : 지정한 지령이름과 관련된 정보를 보여준다.
-G : 지정한 RGID와 관련된 정보를 보여준다.
-U : 지정한 RUID와 관련된 정보를 보여준다.
-g : 지정한 대화접속 혹은 그룹이름과 관련된 정보를 보여준다.
-p : PID를 선택한다.
-s : 대화접속에 속한 프로세스를 선택한다.
-t : tty를 선택한다.
-u : 사용자ID를 지정한다.
U : 지정한 사용자의 프로세스를 보여준다.
p : 프로세스ID를 선택한다.
t : tty를 선택한다.
--Group : 실제그룹이름이나 ID를 선택한다.
--User : 실제사용자이름이나 ID를 선택한다.
--group : 유효사용자이름이나 ID를 선택한다.
--pid : 프로세스ID를 선택한다.
--sid : 대화접속ID를 선택한다.
--tty : 말단을 선택한다.
--user : 유효사용자이름이나 ID를 선택한다.
-123 : --sid와 같다.
123 : --pid와 같다.

결과로서 표시되는 마당을 선택(조종)하는 추가선택
-O : PID, TTY, STAT, TIME, COMMAND 등의 마당목록을 보여준다.
-c : PID, CLS, PRI, TTY, TIME, CMD 등의 마당목록을 보여준다.
-f : UID, PID, PPID, C, STIME, TTY, TIME, CMD 등의 마당목록으로 CMD마당의 전체 지령 형태를 보여준다.
-j : PD, PGID, SID, TTY, TIME, CMD 등의 마당목록을 보여준다.
-l : F, S, UID, PID, PPID, C, PRI, NI, ADDR, SZ, WCHAN, TTY, TIME, CMD 등의 마당목록으로 상세한 정보를 보여준다.
-o : 사용자지정형식
-y : -l 이나 l추가선택과 함께 쓰이여 ADDR마당을 RSS마당으로 보여준다.
--format : 사용자지정형식

결과로서 보이는 마당의 내용을 변경하는 추가선택
-H : 프로쎄스를 계층구조로 보여준다.
-m : 스레드를 보여준다.
-n : 지정한 체계이름 목록파일을 사용한다.
-w : 마당의 너비에 맞추어 자세하게 보여준다.
--cols, --columns : 화면의 너비를 설정한다.
--cumulative : 완료된 자식프로쎄스자료를 포함한다.
--forest : ASCII Art의 프로쎄스나무구조를 보여준다.
--html : HTML탈퇴렬로 보여준다.
--headers : 머리부행을 반복한다.
--no-headers : 머리부를 보이지 않게 한다.
--lines, --rows : 화면의 높이를 설정한다.
--sort=[+|-]key[,[+|-]key[,...] : 내용을 정렬시키는 형식이다. 여기서 사용할수 있는 key는 사용법에서 설명한다.

프로그람의 정보
-V, --version : 판본정보를 보여준다.
--help : 도움말을 보여준다.
--info : 오유제거정보를 보여준다.

ps지령은 프로쎄스의 현재상태를 보여주는 지령이다.

현재 사용하는 프로쎄스의 상태를 간단히 살펴보자. ps지령을 실행하였을 때 나타나는 PID, TTY, TIME, CMD 등의 마당들에 대한 자세한 내용을 지도서페지를 참고하기 바란다.

-u추가선택으로 사용자의 ID나 이름을 지정하여 지정한 사용자의 프로쎄스만을 살펴 볼수 있다.

```
[root@Dolphin root]# ps -u apache -l
F S   UID   PID  PPID  C PRI  NI ADDR    SZ WCHAN  TTY      TIME CMD
5 S    48  2890  2861  0  75   0    -  4961 semop  ?    00:00:00 httpd
5 S    48  2891  2861  0  75   0    -  4961 semop  ?    00:00:00 httpd
5 S    48  2892  2861  0  75   0    -  4961 semop  ?    00:00:00 httpd
5 S    48  2893  2861  0  75   0    -  4961 schedu ?    00:00:00 httpd
5 S    48  2894  2861  0  76   0    -  4961 semop  ?    00:00:00 httpd
5 S    48  2895  2861  0  75   0    -  4961 semop  ?    00:00:00 httpd
5 S    48  2896  2861  0  75   0    -  4961 semop  ?    00:00:00 httpd
5 S    48  2897  2861  0  75   0    -  4961 semop  ?    00:00:00 httpd
```

apache관련프로쎄스중 PID 2891을 강제로 완료시키자. 여기서 -9는 해당 프로쎄스를 강제로 완료시킨다.

```
# kill -9 2891
```

관련 PID프로쎄스를 다시 시작할수도 있다.

```
# kill -HUP 2891
```

또는 다음의 방법을 사용할수도 있다.

```
# kill -SIGHUP 2891
```

보통 체계의 모든 프로쎄스를 보려면 ps와 함께 aux추가선택을 사용한다.

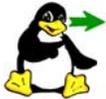
```
[root@Dolphin root]# ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  1372   80 ?        S    21:13   0:04 init
root         2  0.0  0.0     0    0 ?        SW   21:13   0:00 [keventd]
root         3  0.0  0.0     0    0 ?        SW   21:13   0:00 [kapmd]
root         4  0.0  0.0     0    0 ?        SWN  21:13   0:00 [ksoftirqd_CPU0]
root         9  0.0  0.0     0    0 ?        SW   21:13   0:00 [bdflush]
root         5  0.0  0.0     0    0 ?        SW   21:13   0:00 [kswapd]
root         6  0.0  0.0     0    0 ?        SW   21:13   0:00 [kscand/DMA]
root         7  0.0  0.0     0    0 ?        SW   21:13   0:01 [kscand/Normal]
```

프로쎄스들을 어떤 기준으로 정렬해야 할 경우가 많다. 정렬을 위해서는 O추가선택이나 —sort추가선택을 사용한다. O추가선택일 경우 O[+|-]K[[+|-]K[,...]의 형식을 사용하며 —sort추가선택일 경우 —sort=[+|-]key[,[+|-]key[,...]의 형식을 사용한다. O추가선택에 -r를 붙여 RSS를 가장 큰 값으로부터 보여주는 내림차순으로 정렬하였다. 만일 두개이상의 기준으로 정렬을 한다면 반점뒤에 정렬방식과 기준되는 K마당을 지정하면 된다.

여기에서는 O추가선택이나 —sort에서 사용가능한 key를 보여준다. K마당은 O추가선택에, key마당은 —sort추가선택에 사용한다.

K	key	설명
c	cmd	실행이름
C	cmdline	전체 지령행
f	flags	기발
g	pgrp	프로쎄스그룹ID
G	tpgid	tty에 속한 프로쎄스그룹ID
j	cutime	루적사용자시간
J	cstime	루적체계시간
k	utime	사용자시간
K	stime	체계시간
m	min_flt	보다 작은 페지기정값
M	ma_flt	보다 큰 페지기정값
n	cmin_flt	보다 작은 페지기정값의 루적수

N	cmaj_flt	보다 큰 페지기정값의 루적수
o	session	대화접속ID
p	pid	프로쎄스ID
P	ppid	부모프로쎄스ID
r	rss	상주기억기의 크기
R	resident	상주페지
s	size	사용기억기크기(KB단위)
S	share	공유페지의 수
t	tty	tty
T	start_time	프로쎄스가 시작된 시간
U	uid	사용자ID
u	user	사용자이름
v	vsize	전체 VM크기(B단위)

련관지령

pstree 프로쎄스계층구조를 나무구조형태로 보여준다.

top 실시간으로 프로쎄스변화상태를 보여준다.

pstree

/usr/bin/pstree

프로쎄스상관도

지령형식: pstree [추가선택] [pid | user]

pstree -V

-a : 지령행에서 지정한 인수가 있다면 지령행인수까지 보여준다.
-c : 나무구조에서 같은 프로쎄스를 모두 보여준다.
-G : VT100형태로 보여준다.
-h : 현재프로쎄스와 부모프로쎄스를 밝은색으로 보여준다.
-H pid : pid로 지정된 프로쎄스와 부모프로쎄스를 밝은색으로 보여준다.
-l : 긴 행을 모두 보여준다.
-n : PID를 기준으로 정렬하여 보여준다.
-p : PID까지 보여준다.

-u : UID를 보여준다.
-U : UTF-8형태로 보여준다.
-V : 판본정보를 출력한다.
pid : 지정하는 pid를 보여준다.
user : 지정하는 사용자만의 프로쎄스를 보여준다.

pstree지령은 프로쎄스사이의 부모자식관계를 나무구조형태로 보여주어 리해하기 쉽게 해준다. 우의 화면에서 script는 gimp의 자식프로쎄스임을 알수 있다. 또한 모든 프로쎄스는 init의 자식프로쎄스로 련결되여있다.

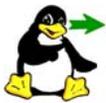
pwck /usr/sbin/pwck

통과어파일검사

 지령형식: pwck

pwck지령은 /etc/passwd파일의 내용을 분석하여 잘못된 내용이 있는가를 검사하는 지령이다. 통과어의 보안문제점을 미리 발견하는 지령으로서 정기적으로 검사해보아야 한다.

아래와 같이 하면 /etc/passwd파일의 문제점을 자동적으로 검사해준다.

/etc/passwd파일에서 adm계정은 home등록부를 /var/adm으로 지정하였지만 실제로는 지정된 등록부가 없으므로 오유통보문을 내보낸다. 이 계정들은 /etc/passwd파일에서 삭제하거나 주석(#)처리하는것이 좋다.

련관지령

grpck　　/etc/group파일검사지령

Pwconv

/usr/sbin/pwconv

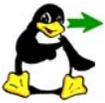
그림자통과어 사용

지령형식: pwconv

통과어파일을 그림자로 변환하며 통과어운영에서도 그림자방식을 사용한다.

/etc/passwd파일은 일반사용자도 읽을수 있으므로 보안상 결함을 가지고있다. pwconv지령은 /etc/passwd에 있는 자료를 변경하고 현재 존재하고있는 그림자정보에 포함시킨다. 통과어가 설정되지 않은 사용자는 /etc/shadow에 추가시키지 않으며 통과어의 기한이 끝나거나 경고잠금 등의 정보를 포함하고있는 /etc/login.defs에서 정의한 값을 사용하게 된다. 최근 배포판에는 pwconv가 설정되여있으므로 특별히 설정할 필요가 없다.

이 설정을 해제하려면 pwunconv지령을 사용한다. 이것은 그림자자료를 /etc/passwd형식으로 변경하고 /etc/shadow파일을 삭제하므로 사용하지 않는것이 좋다.

/etc/shadow파일은 우의 보는바와 같이 root만이 읽을수 있으므로 파괴를 막을 수 있다.

련관지령

grpconv 그림자그룹을 만드는 지령

pwd

/bin/pwd

현재경로보기

지령형식: pwd

현재 사용하는 등록부의 절대경로를 보여주는 지령이다. 실례로 /etc/httpd/conf등록부로 이동한 후 작업을 하는 도중에 현재등록부를 보려면 pwd지령을 리용한다.

참고로 임의의 작업등록부에서 cd지령을 입력하면 $HOME등록부로 이동한다.

quota

/usr/sbin/quota

디스크사용용량 제한

지령형식: quota [추가선택] [사용자 | 그룹]

-g : 계정과 관련된 그룹의 용량제한정보를 보여준다.
-q : 용량제한을 넘긴 사용자의 파일체계정보를 보여준다.
-u : 기정추가선택이다. 사용자의 용량제한정보를 보여준다.
-v : 보다 구체적인 용량제한정보를 보여준다.

사용자가 사용하는 디스크의 크기를 제한하여 체계관리의 효률성을 높이게 하는 지령이다. quota지령으로 등록가입한 계정의 quota설정내용을 볼수 있다.

용량제한의 설정(Red Hat 7.X이상)

1) /etc/fstab에서 용량제한을 하고싶은 파일체계에 usrquota를 입력한 후 재기동한다.

```
LABEL=/ /ext2 defaults, usrquota 1 1
```

2) 우에서의 설정은 /에 제한하였으므로 /등록부에 touch지령으로 quota.user파일을 생성한다.

```
# touch/quota.user

# chmod 600 /quota.user
```

3) convertquota지령으로 quota.user를 aquota.user자료파일로 생성한다.

```
# convertquota /
```

4) edquota admin지령으로 admin사용자의 soft와 hard분담몫을 설정한다.

5) repquota지령으로 파일체계의 분담몫을 간략해서 볼수 있다.

```
[root@Dolphin root]# repquota -a
*** Report for user quotas on devicd /dev/hda6 (/)
Block grace time:7 days; Inode grace time: 7 days
Block limits File limits
User used soft hard grace used soft hard grace
..........................
root - 42668 0 0 28 0 0
admin - 484 0 0 118 0 0
```

련관지령

edquota	용량제한을 위해 매 설정을 편집하는 지령
quotacheck	용량제한설정을 검사하는 지령
quotaoff	용량제한을 해제하는 지령
quotaon	용량제한을 하는 지령
quotastats	용량제한상태를 보는 지령

ramsize

/usr/bin/ramsize

RAM디스크 크기

지령형식: ramsize [추가선택] [image[size[offset]]]

-o offset : 지정한 offset로 offset값을 지정한다.

RAM디스크의 크기를 보여주거나 설정하는 지령이다.
ramsize지령으로 핵심부사본을 지정하면 그것의 offset정보를 보여준다.

ramsize지령은 아래에서 보는바와 같이 rdev지령에 기호련결되여있다.

자세한 내용은 rdev지령을 참조하기 바란다.

RAM disk

하드디스크를 흉내내여 기억기에 상주하는 프로그람이다. 자료를 저장하기 위해 RAM의 일부를 사용한다. RAM disk는 하드디스크에 비해 자료호출속도가 훨씬 빠르지만 전원이 꺼지면 자료가 지워지게 된다.

rcp

/usr/bin/rcp

원격파일복사

지령형식: rcp [추가선택] 파일1 파일2
rcp [추가선택] 파일... 등록부

-p : 원본파일의 변경날자와 방식을 그대로 복사한다.
-x : 콤퓨터들사이의 모든 정보를 DES암호화한다.
-r : 등록부안의 부분등록부와 파일들을 모두 복사한다.
-D port : 지정한 포구로 접속한다.

rcp지령은 망에 련결되여있는 매 콤퓨터사이에 원격복사(remote copy)로 자료이동을 실현시킨다. r로 시작하는 지령들은 대부분 보안측면에서 부족점들이 있으므로 rcp대신 scp지령을 사용하는것이 더 안전하다.

DES (Data Encryption Standard): 자료암호화표준

매 64bit 자료블로크에 56bit길이의 열쇠를 적용하는 암호화방식이다. 이 과정은 여러가지 방식으로 실행될수 있으며 16번의 연산이 반복된다. 비록 DES가 강력한 암호화로 인정되고는 있지만 대부분 세개의 열쇠가 적용되는 "3중DES"를 사용한다. 그렇다고 해서 DES로 암호화된 통보문이 해독될수 없다는것은 아니다.

rdate

/usr/bin/rdate

시간맞추기

지령형식: rdate [추가선택] [주콤퓨터...]

-p : 주콤퓨터의 자료를 보여준다.
-s : 원격망콤퓨터로부터 콤퓨터시간을 설정한다.

시간봉사기의 시간을 참고하여 체계의 시간을 변경하는 지령이다.
rdate -p추가선택으로 time.sec.net 봉사기의 시간을 확인하자.

rdate -s추가선택으로 국부체계의 시간을 시간봉사기의 시간으로 바꿀수 있다.

```
# rdate -s time.sec.net
```

date지령으로 변경된 시간을 확인할수 있다.

주의 보통 지령행에서 입력하는 값은 체계를 재기동하면 모두 없어진다. 기동할 때마다 실행되여야 하는 지령들은 /etc/rc.d/rc.local파일에 추가하여야 한다. 우에 입력한 rdate -s time.sec.net지령도 역시 /etc/rc.d/ rc.local파일에 추가하여 체계를 기동할 때마다 시간봉사기와 시간을 맞추게 할수 있다.

rdev

/usr/sbin/rdev

장치검사

지령형식: rdev [추가선택] [-o offset] [image [value [offset]]]
rdev [-o offset] [image [root-device [offset]]]

-r : ramsize지령과 같다.
-R : rootflags지령과 같다.
-v : vidmode지령과 같다.
-h : 도움말을 보여준다.

rdev는 RAM디스크의 크기, root기발, 화면방식의 상태를 보여주고 설정하는 지령이다. 이에 대한 정보는 체계기동에 사용되는 기동사본안에 들어있다.

먼저 RAM디스크의 크기를 확인해보자. RAM디스크의 크기는 KB단위이다.

```
root@Dolphin:~
[root@Dolphin root]# rdev -r /boot/vmlinuz
Ramsize 0
```

현재 RAM디스크로 사용되는 용량은 없다. rdev -r은 ramsize와 같은 지령으로서 ramsize은 rdev와 기호련결되여있다.

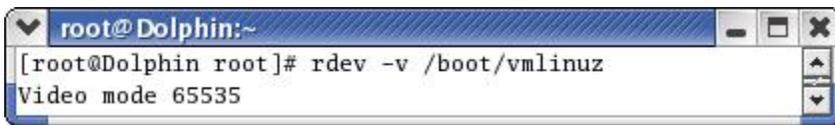
```
root@Dolphin:~
[root@Dolphin root]# ls -al /usr/sbin/ramsize
lrwxrwxrwx    1 root     root            4 Feb 13 07:19
/usr/sbin/ramsize -> rdev
```

다음 root기발을 확인해보자.

rdev -R지령에서 사용되는 기발인수값은 root파일체계를 탑재할 때 추가적인 정보를

담고있는 값이다. 일반적으로 기발인수값이 0이 아니라면 핵심부가 root파일체계를 읽기 전용으로 탑재하도록 한다.

rdev -R지령도 rootflags와 같은 지령으로서 rootflags는 rdev와 기호련결되여있다.

마지막으로 화면방식을 확인해보자.

rdev -v지령은 vidmode지령과 같으며 vidmode지령은 rdev와 기호련결되여있다. 이 지령에서 사용되는 mode값은 다음과 같은 화면방식을 지정한다.

- -3 입력재촉문방식
- -2 확장 VGA방식
- -1 표준 VGA방식
- 0 입력재촉상태에서 "0"이 눌리워진것으로 본다.
- 1 입력재촉상태에서 "1"이 눌리워진것으로 본다.
- 2 입력재촉상태에서 "2"가 눌리워진것으로 본다.
- n 입력재촉상태에서 "n"이 눌리워진것으로 본다.

rdev지령에서 value값을 지정하지 않으면 image값은 단지 현재설정을 알기 위해서 쓴다.

Apache, PHP, MySQL설치

Apache, PHP, MySQL을 설치하면 설치원본들을 /usr/local등록부에 복사하고 압축을 푼다.

다음과 같이 설치하면 쉬울것이다. 매 ./configure부분에 추가한 설정도 일반적으로 사용하는것이다.

1. /usr/local등록부에 Apache, PHP, MySQL원본들을 풀어놓는다.
2. /usr/local/mysql등록부로 이동하여 MySQL설치를 시작한다.

```
# ./configure --prefix=/usr/local/mysql \ >
--localstatedir=/usr/local/mysql --with-charset=euc_kp
# make;make install;
# cd /usr/local/mysql
# ./bin/mysql_install_db
# ./bin/safe_mysqld &
```

3. /usr/local/apache등록부로 이동하여 Apache경로를 설정한다.

```
# ./configure --prefix=/usr/local/apache
```

4. /usr/local/php등록부로 이동하여 PHP설치를 한다.

```
# ./configure --with-apache=./apache \ >
--with--mysql=/usr/local/mysql -enable-track-vars=yes
# make;make install;
```

5. 이제는 Apache를 설치한다.

```
# cd /usr/local/apache/src
# ./configure
# cd..
# ./configure --prefix=/usr/local/apache \ >
--activate-module=src/modules/php4/libphp4.a
# make;make install;
# cd /usr/local/apache/bin
# ./apachectl start
```

readonly

/usr/bin/readonly

읽기전용

지령형식: readonly [추가선택] [이름...]

-f : 이름에 해당하는 함수를 읽기전용으로 표기한다.
-P : 모든 읽기전용변수목록을 보여준다.

주어진 이름들을 읽기전용으로 표기하고 다음에 나올 대입문에 의해 이름들의 값이 바뀌지 않도록 해주는 지령이다. --인수는 나머지 인수에 대한 검사을 하지 않도록 지시한다. 잘못된 추가선택이 있거나 이름들중 하나라도 적합한 쉘변수이름이 아닌 경우 또는 -f 다음에 나온 이름이 함수가 아닌 경우를 제외하고는 반환상태값이 0이다. 이에 대해서는 쉘스크립트부분을 참고하기 바란다.

reboot

/usr/bin/reboot

체계재기동

지령형식: reboot [추가선택]

-d : wtmp파일에 기록을 남기지 않는다.
-f : 강제로 재기동한다.
-n : 재기동하거나 체계정지하기전에 동기화하지 않는다.
-w : 재기동하지 않고 /var/log/wtmp파일에 기록을 남긴다.
-i : 재기동하거나 체계정지하기 직전에 모든 망대면부들을 닫는다.
-h : 체계를 정지하거나 완료하기 직전에 체계의 모든 하드웨어장치들을 실행대기방식으로 놓는다.
-p : 체계를 정지할 때 전원을 끈다.

체계관리자외 다른 일반사용자가 체계를 사용하지 않을 때 체계를 다시 기동하는 지령이다.

이 지령은 shutdown -r now지령이나 init 6지령과 같다. shutdown지령은 보통 체계가 수행중에 있거나 사용자가 현재 작업하는것을 보관할 시간적여유를 주는 경우 통보문을 내보낸다. 그러나 reboot지령은 경고없이 즉시에 체계를 재기동하게 된다.

reboot지령은 syslog지령을 실행한 결과를 /var/log/message와 등록가입자통계기록파일인 /var/log/wtmp에 보관하게 한다.

renice /usr/bin/renice

우선순위변경

지령형식: renice 우선권 [추가선택] [프로세스]

+num : 현재프로세스의 우선순위를 지정한 수만큼 낮춘다.
-num : 현재프로세스의 우선순위를 지정한 수만큼 높여준다.
-g : 그룹ID 지정
-u : 사용자ID 지정
-P : 프로세스ID 지정(초기값)

실행중에 있는 프로세스들의 작업우선권을 변경하는 지령이다. 이 지령을 nice지령과 비교하여 보는것이 좋다.

renice에서 사용되는 추가선택은 프로세스ID, 프로세스그룹ID, 사용자ID이다. 프로세스그룹의 우선권변환은 그 그룹에 속하는 모든 프로세스에 다 적용된다. 사용자의 우선권변환은 그 사용자가 진행하는 모든 프로세스에 적용된다.

nice지령에서 renice지령을 함께 실행해보았으므로 그것을 참고하기 바란다.

reset /usr/bin/reset

말단기초기화

지령형식: reset [추가선택] [말단기]

- : 현재말단기의 종류를 보여준다.
-e ch : 지정된 ch문자를 말단기에서 ERASE문자로 설정한다.
-i ch : -e추가선택과 류사하지만 지정된 ch문자를 말단기에서 새치기문자로 설정한다.
-k ch : 지정한 ch문자를 말단기에서 LINE KILL문자로 설정한다.
-m mapping : 말단기에서의 포구형태로 지정한다.

말단기(Terminal)를 초기화하는 지령이다.

clear지령은 말단기화면상에서 깨끗하게 지운 다음 유표를 화면우로 이동시키는 지령이지만 reset는 말단기를 초기화한다.

프로그람이 완료된 상태에서 그 어떤 리유로 일반적인 지령이 실행되지 않는 경우 Ctrl+j clear Ctrl+j지령으로 말단기를 초기화할수도 있다.

reset지령은 tset지령의 기호련결파일이다.

련관지령

clear 화면을 깨끗이 하고 유표를 화면우로 이동시킨다.

restore

/sbin/restore

dump여벌복사파일의 복구

지령형식: restore [추가선택]

```
restore -C [-ckMvy] [-b blocksize] [-D filesystem] [-f file] [-s fileno]
    [-T directory]
restore -i [-chkmMNuvy] [-b blocksize] [-f file] [-Q file] [-s fileno]
    [-T directory]
restore -R [-ckMNuvy] [-b blocksize] [-f file [-s fileno] [-T directory]
restore -r [-ckMNuvy] [-b blocksize] [-f file] [-s fileno] [-T directory]
restore -t [-chkMNuvy] [-b blocksize] [-f file] [-Q file] [-s fileno]
    [-T dilectory] [-X filelist] [filename]
restore -X [-chkmMNuvy] [-b blocksize] [-f file] [-f file] [-s fileno]
    [-T directory] [-X filelist] [filename]
```

dump여벌복사파일을 복구한다.

dump지령은 복잡한 파일체계의 여벌복사편의프로그람으로서 보통 규모가 큰 설치와 망에서 쓰인다. restore지령은 dump지령으로 여벌복사된 파일을 복구한다.

rev

/usr/bin/rev

내용 거꾸로 보기

지령형식: rev [파일]

파일의 내용을 렬단위로 영어자모순서를 거꾸로 바꾸어 라렬하는 지령이다.

rev지령은 지정한 파일을 렬단위로 읽어 그 렬의 내용을 역변환하여 보여준다. 파일이 지정되지 않으면 표준입력에서 받는다.

text라는 파일을 한번 살펴보자.

```
root@Dolphin:~
[root@Dolphin root]# cat text
Linux version 2.4.20-8 (bhcompile@porky.devel.redhat.com) (g
cc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5)) #1 Thu Ma
r 13 17:54:28 EST 2003
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
 BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000e0000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000000f7f0000 (usable)
 BIOS-e820: 000000000f7f0000 - 000000000f800000 (reserved)
 BIOS-e820: 00000000fec00000 - 0000000100000000 (reserved)
```

rev지령은 text파일의 매 렬의 내용을 거꾸로 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# rev text
3002 TSE 82:45:71 31 raM uhT 1# ))5-2.2.3 xuniL taH deR( 222
03002 2.2.3 noisrev ccg( )moc.tahder.leved.ykrop@elipmochb(
8-02.4.2 noisrev xuniL
:pam MAR lacisyhp dedivorp-SOIB
)elbasu( 00cf900000000000 - 0000000000000000 :028e-SOIB
)devreser( 0000a00000000000 - 00cf900000000000 :028e-SOIB
)devreser( 0000010000000000 - 0000e00000000000 :028e-SOIB
)elbasu( 0000f7f000000000 - 0000010000000000 :028e-SOIB
)devreser( 000008f000000000 - 0000f7f000000000 :028e-SOIB
)devreser( 0000000010000000 - 00000cef00000000 :028e-SOIB
```

rexec

/usr/bin/rexec

원격실행

지령형식: rexec [추가선택] -l username -p password host command

-l username : 원격콤퓨터에 접속할 사용자이름을 지정한다.
-p password : 원격콤퓨터에 접속할 사용자이름의 통과어를 지정한다.
host : 접속할 원격콤퓨터이름
command : 원격콤퓨터에 접속하여 실행할 지령을 지정한다.
-n : 접속할 사용자와 통과어입력대기를 보여준다.
-h : 사용법을 보여준다.

rexec지령은 원격콤퓨터에서 실행할 지령을 입력하여 지령의 결과를 국부체계에서 볼수 있게 하는 지령이다. r로 시작하는 지령은 대부분 보안상 결함이 있다는데 대하여 주의해야 한다.

rexec지령의 사용법에 대해 간단히 보기로 하자.

linuxroot.co.kp콤퓨터에서 date지령을 실행하여 이 콤퓨터의 시간을 한번 알아보자.

```
# rexec linuxroot.co.kp date
```

우와 같이 입력하면 국부체계의 $HOME/.netrc라는 파일에 방금 접속하였던 원격콤퓨터의 사용자와 통과어를 저장하게 된다.

아래의 실례는 linuxroot.co.kp콤퓨터에 있는 admin계정의 home등록부목록을 국부체계에서 보여준다.

```
# rexec linuxroot.co.kp ls -l /home/admin
```

주의 rexec를 사용하기 위해서는 원격콤퓨터가 rexec봉사를 지원해야 한다. rexec봉사는 슈퍼데몬인 xinitd에 포함되여있으며 /etc/xinitd.d/rexec파일을 변경하여 사용할수 있다.

rlogin

/usr/bin/rlogin

원격콤퓨터접속

지령형식: rlogin 콤퓨터 [추가선택]

-t ttytype : tty형태를 지정한다.
-l username : 접속할 사용자계정을 지정한다.

자주 접속하는 콤퓨터에 자동으로 telnet접속을 할수 있게 허용하는 지령이다.

rlogin은 telnet과 비슷한 기능을 수행한다. 하지만 rlogin을 리용하면 허용된 콤퓨터들에 자동적으로 등록가입할수 있다.

실례로 만일 linux1.linuxroot.co.kp에서 linux2.linuxroot.co.kp로 telnet등록가입하려면 매번 계정을 입력한 다음 통과어를 입력해야 한다. 하지만 rlogin을 사용하면 linux1.linuxroot.co.kp콤퓨터를 자동으로 등록가입하도록 할수 있다.

rlogin지령도 보안상 결함을 가지고있으므로 rlogin지령보다 ssh지령을 사용하는것이 더 좋을것이다.

```
root@Dolphin:~
[root@Dolphin root]# cat /etc/xinetd.d/rlogin
# default: on
# description: rlogind is the server for the rlogin(1) progr
am.  The server \
#       provides a remote login facility with authentication
 based on \
#       privileged port numbers from trusted hosts.
service login
{
        socket_type             = stream
        wait                    = no
        user                    = root
        log_on_success          += USERID
        log_on_failure          += USERID
        server                  = /usr/sbin/in.rlogind
        disable                 = yes
}
```

련관지령

telnet　　원격 Telnet봉사기에 접속하기 위한 의뢰기

ssh　　원격 SSH 봉사기접속의뢰기

/bin/rm

파일삭제

지령형식: rm [추가선택] 파일이름

-d, —directory : rmdir지령대신에 unlink와 함께 경로를 지운다.
-f, --force : 지울 파일이 없을 경우 아무런 통보문도 보여주지 않고 그냥 넘어간다.
-I, —interactive : 삭제하기 전에 사용자에게 삭제여부를 확인한다.
-r, -R, --recursive : 부분등록부를 포함한 모든 내용을 삭제한다.
-v, —verbose : 지우는 파일의 정보를 화면에 출력한다.
-R : -r추가선택과 같은 역할을 한다.
—help : 도움말을 보여준다
--version : 판본정보를 보여준다.

rm지령은 등록부나 파일을 삭제하는 지령이다.

보통 배포판에는 $HOME/.bashrc파일에 alias rm='rm -i'를 설정하여 조작실수로 파일을 삭제하는 경우 삭제여부를 확인하게 한다.

만일 삭제할 파일이나 등록부가 정확히 확인된것이라면 -rf추가선택으로 등록부를 포함한 부분등록부와 파일들을 모두 삭제할수 있다.

실례를 들어보자.

이 등록부를 삭제해보자.

```
# rm -rf php-4.3.0-Win32
```

rm -ir추가선택으로 yes/no를 선택하여 부분등록부들가운데서 지정된 파일만 삭제할 수도 있다.

주의 추가선택 다음에 오는 --은 추가선택이 아닌것으로 인식한다. 즉 "-f"라는 이름을 가진 파일을 지우려고 한다면 다음의 두가지 방법을 사용한다.

rm -- -f 혹은 # rm ./-f

련관지령

rmdir 등록부삭제지령

/usr/bin/rmail

원격우편보기

지령형식: rmail [추가선택] 사용자

-D domain : 기정값인 UUCP대신에 지정한 령역을 설정한다.
-T : 오유제거방식

UUCP를 통하여 받은 원격우편을 다루는 지령이다.

UUCP환경에서는 우편을 주로 rmail지령으로 전송한다. rmail지령은 통보문을 다른 콤퓨터에로 전송하며 그 통보문은 목적콤퓨터에 도착하게 된다.

현재는 거의 쓰지 않는다.

UUCP (UNIX-to-UNIX Copy Protocol)

UUCP는 서로 다른 콤퓨터들에 설치되여있는 UNIX체계들사이에 파일을 복사하고 다른 체계상에서 실행할 지령들을 보내기 위한 UNIX프로그람묶음이다. UUCP의 주요 지령들은 다음과 같다.

uucp 특정한 파일을 지정한 다른 체계로 복사할것을 요구한다.
uux 다른 체계에서 실행할 UNIX지령을 보낸다.
uucico 복사를 진행하고 실행하는 프로그람으로서 UNIX체계상에서 실행된다. 일반적으로 이 프로그람은 하루에 여러번 실행되는데 그동안에 복사(uucp) 및 지령요청(uux)들은 uucico프로그람이 실행될 때까지 대기렬에 저장된다.
uuxqt uucico프로그람에 의하여 시작된 후 uux가 보낸 지령들을 수행한다.

/bin/rmdir

등록부삭제

지령형식: rmdir [추가선택] 등록부

-p, --parents : 부모등록부도 삭제한다. 그 부모등록부도 비여있는 경우에만 실행할수 있다.
--verbose : 진행상태를 구체적으로 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

부분등록부나 파일이 없는 등록부를 삭제하는 지령이다. 만일 부분등록부나 파일이 있는 등록부를 삭제하려면 rm -r지령을 사용한다.

rmdir지령은 등록부가 비여있지 않으면 삭제되지 않는다. 비여있지 않는 php-4.3.0-Win32등록부를 한번 삭제해보자.

먼저 부분등록부의 파일이나 등록부를 rm지령으로 삭제해야 한다.

련관지령

rm 파일이나 등록부삭제지령

rmmod

/sbin/rmmod

모듈해제

지령형식: rmmod [추가선택] 모듈이름

-a, --all : modprobe -a로 자동적재된 모듈까지 해제한다.
-h, --help : 도움말을 보여준다.
-r, --stacks : 모듈탄창을 제거한다.
-s, —syslog : syslog에 통보문을 남긴다.
-v, --verbose : 진행상태를 구체적으로 보여준다.
-V, --version : 판본정보를 보여준다.

rmmod지령은 체계에 적재되여있는 모듈을 해제할 경우 사용하는 지령이다. 이것은 아래의 **련관지령**들과 매우 밀접한 련관이 있으므로 매 지령들을 한번씩 련관시켜서 살펴보는것이 좋다.

현재 체계에 적재되여있는 모듈들은 lsmod지령으로 확인할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# lsmod
Module                  Size  Used by    Not tainted
ide-cd                 35708   0  (autoclean)
cdrom                  33728   0  (autoclean) [ide-cd]
agpgart                47776   5  (autoclean)
parport_pc             19076   1  (autoclean)
lp                      8996   0  (autoclean)
parport                37056   1  (autoclean) [parport_pc lp
autofs                 13268   0  (autoclean) (unused)
e100                   60644   1
microcode               4668   0  (autoclean)
nls_iso8859-1           3516   2  (autoclean)
nls_cp437               5116   2  (autoclean)
vfat                   13004   2  (autoclean)
fat                    38808   0  (autoclean) [vfat]
keybdev                 2944   0  (unused)
```

그중에서 keybdev모듈을 삭제해보자. 아무 오유가 없으면 성공이라고 볼수 있다.

```
# rmmod keybdev
```

련관지령

depmod　　모듈적재를 위한 의존성관계를 관리하는 지령
modprobe　　insmod보다 우월한 모듈적재지령
insmod　　모듈적재지령

rootflags

/usr/sbin/rootflags

root장치설정

ROOT파일체계를 탑재한다.

rootflags지령은 실제로는 rev -R지령이며 이 지령과 기호련결되여있다. 자세한 내용은 rdev지령을 보기 바란다.

route

/sbin/route

경로배정표보기

지령형식: route [추가선택] [지령]

추가선택
-A family : 지정한 주소를 설명해준다.
-v, --verbose : 보다 구체적인 정보를 보여준다.
-n, —numeric : Destination마당값을 모두 IP주소로 보여준다.
-e, —extend : 좀 더 자세한 경로배정표정보를 보여준다.
-F, --fib : 기본적인 경로배정정보를 보여준다.
-C, --cache : 경로배정완충정보를 보여준다.
-h, --help : 도움말을 보여준다.
-V, --version : 판본정보를 보여준다.

지령
add [-net|-host] IP주소 [gw gateway] [netmask netmask] [mss tcp-mss] [dev device]
del IP주소

IP경로배정표를 보여주거나 관리하는 지령이다.

route지령은 망경로배정표에 수동항목을 작성할수 있다. /etc/hosts에 설정한 콤퓨터별명이나 /etc/resolve.conf파일에 설정된 이름봉사기의 정보에서 주소를 해석하게 된다.

```
[root@Dolphin root]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.8.0     *               255.255.255.0   U     0      0        0 eth0
169.254.0.0     *               255.255.0.0     U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         Dolphin.co.kp   0.0.0.0         UG    0      0        0 eth0
```

보는바와 같이 eth0망장치는 192.168.8.0의 망주소를 가지고 Genmask 즉 망마스크는 255.255.255.0의 값을 가진다. eth0망장치는 또한 169.254.0.0의 망주소를 가지고 255.255.255.0의 망마스크값을 가진다. lo장치는 망기판이 인식되여있다면 자동적으로 인식하는 장치로서 자기의 IP 즉 국부콤퓨터주소를 가지고있는 127.0.0.1에 대한 장치경로배정표이다. 경로배정표중에서 Destination마당에 있는 기정값은 관문주소를 의미하는것으로서 169.254.0.1의 값을 가지고있다.

-n추가선택은 콤퓨터이름대신 4분표기법으로 된 주소를 나타낸다.

```
[root@Dolphin root]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.8.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         192.168.8.83    0.0.0.0         UG    0      0        0 eth0
```

경로배정표에 나타난 기본관문을 한번 삭제해보자. 이것은 route del지령으로 Destionation값이 기정인 관문을 삭제하므로 gw나 gateway를 입력하며 관문주소를 지정하면 된다.

```
# route del default gw 192.168.0.1
```

기본관문의 추가는 route add 지령으로 진행한다.

```
# route add default gw 169.254.0.1
```

경로기(router)

경로기는 서로 떨어져있으면서 같은 전송규약을 사용하는 망들을 련결해주는 장치이다. 경로기는 망다리가 가지고있는 기능을 확장하여 경로배정표에 따라 다른 망 또는 자기 망의 마디를 결정해준다. 그리고 여러 경로중에서 가장 효률적인 경로를 선택하여 파케트를 보낸다. 경로기는 자료흐름을 조종하며 내부망에서 여러 보조망을 구성하고 다양한 망관리기능을 수행한다. 망다리와 경로기의 차이점을 간단히 보면 경로기는 ISO참조모형에서 망층까지의 기능을 담당하고있으면서 경로설정을 해주며 망다리는 자료련결층까지의 기능만으로 목적지주소에 따르는 파케트의 선별 및 경로결정을 한다.

관문(gateway)

관문은 다른 망으로 들어가는 입구역할을 하는 망마디이다. 경로배정의 관점에서 보면 Internet는 많은 관문과 콤퓨터마디들로 구성된 망이라고 할수 있는데 망사용자들의 콤퓨터들이 바로 콤퓨터마디이며 망안에서 통화량을 조종하는 콤퓨터 또는 Internet봉사제공콤퓨터가 바로 관문이다.

rpm

/bin/rpm

RedHat패키지 관리도구

지령형식: rpm [추가선택]

-w : 구체적인 오유수정정보를 출력한다.
--keep-temps : 림시파일을 지우지 않는다.(/tmp/rpm-*) rpm에 대한 오유수정을 할 때에만 주로 사용한다.
--quiet : 출력을 하지 않고 통보문들만 출력한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.
--rcfile <filename> : /etc/rpmrc 또는 $HOME/rpmrc를 사용하지 않고 지정한 파일을 사용하도록 한다.
--root <dir> : 모든 동작에 대하여 최상위등록부를 주어진 등록부로 설정하고 작업한다. 실례로 Linux를 처음 설치할 때 주어진 등록부를 "/"이라고 생각하고 그것을 기준으로 설치해나간다.

체계에 프로그람을 설치하기 위한 도구로서 대부분의 배포판에서 다 사용할수 있는 패키지관리도구이다.

설치방식	rpm -i[설치추가선택] <패키지파일>+
질문방식	rpm -q[질문추가선택]
검증방식	rpm -V|-y| -verify [검증추가선택]
서명확인방식	rpm --checksig <패키지파일>+
제거방식	rpm -e <패키지파일>+
제작방식	rpm -bO [제작추가선택] <패키지spec>+

매 추가선택에 대하여 구체적으로 보기로 하자.

설치추가선택

rpm 설치지령의 일반적인 형태는 다음과 같다.

```
rpm -i[설치추가선택] <패키지파일>+
```

- force: —replacepkgs, --replacefiles, —oldpackage를 모두 사용한것과 같다.
- h, —hash: 패키지를 풀 때 "#"(Hash표식)를 표시한다. 총 개수는 50개이다. -v를 함께 사용하면 보다 구체적으로 표시한다.
- oldpackage: 새로운 패키지를 지우고 판본이 낮은 패키지로 교체할 때 사용한다.
- percent: 패키지파일을 설치할 때 퍼센트표시를 한다.
- replacefiles: 이미 설치된 다른 패키지의 파일을 덧쓰기하면서 패키지를 강제설치한다.
- replacepkgs: 패키지가 이미 설치되여있다 하더라도 다시 설치한다.
- root <등록부>: 지정한 <등록부>를 뿌리로 하는 체계에 설치한다. 즉 자료기지는 <등록부>를 뿌리로 하여 갱신되며 pre스크립트 또는 post스크립트는 지정한

<등록부>를 파라메터로 하여 chroot()지령을 실행한 후 실행된다.

- noscripts: preinstall, postinstall스크립트를 실행하지 않는다.
- excludedocs: "DOCS"(문서파일)라고 표시되여있는 파일(사용지도서와 texinfo 문서)은 설치하지 않는다.
- includedocs: "DOCS"(문서파일)를 포함한다. 이 추가선택은 rpmrc파일에 excludedocs: 1 이라는 표시가 있을 때에만 필요하다.
- nodeps: 패키지를 설치하기 전에 의존성을 검사하지 않는다.
- test: 패키지를 설치하지 않고 모의설치를 한 다음 검사하고 표시한다.
- -U, —upgrade: 현재 설치되여있는 패키지를 새로운 판본의 RPM으로 갱신한다. -i추가선택과 같지만 낮은 판본의것이 자동으로 지워진다는것이 다르다.

질문추가선택

rpm 질문추가선택의 일반적인 형식은 다음과 같다.

```
rpm -q[질문추가선택]
```

여기에서는 패키지정보가 표시될 형식을 지정해주어야 한다. —queryformat추가선택 뒤에 형식문자렬을 적어주면 된다.

질문형식은 표준 printf형식을 약간 변형한것이다. 형식은 정적문자렬과 printf 형식지정문자(실례로 C에서와 같이 행시작문자, Tab, 다른 특수문자표기)들로 구성되여있다. 출력형태가 지정되면 형식지정문자는 생략하고 머리부태그의 이름을 괄호("{ }")로 묶어서 지정한다. 태그이름중에서 RPMTAG_부분은 생략해야 하며 태그이름앞에는 "-"문자를 적어주어야 한다.

실례로 질문대상패키지의 이름만 출력하려고 하는 경우 %{NAME}을 형식문자렬로 사용해야 한다. 또한 패키지이름과 배포판정보를 두개의 란으로 표시하려고 할 때는 %-30{NAME}%{DISTRIBUTION}이라고 적는다.

rpm지령에 --querytags추가선택을 주면 인식하고있는 모든 태그들의 목록을 보여준다.

질문추가선택에는 패키지선택과 정보선택이 있다.

패키지선택

- <패키지이름>: 지정한 이름을 가진 패키지에 대하여 질문을 수행한다.
- a: 모든 패키지에 대하여 질문을 수행한다.
- whatrequires <기능>: <기능>을 필요로 하여 정확히 동작하는 모든 패키지에 대하여 질문을 수행한다.
- whatprovides <가상>: <가상>기능을 제공하는 모든 패키지에 대하여 질문을 수행한다.

- f <파일>: <파일>을 포함하는 패키지에 대하여 질문을 수행한다.
- F: -f와 같지만 파일이름을 표준입력에서 읽어들인다.
- p <패키지파일>: 설치된 또는 설치되지 않은 <패키지파일>에 대하여 질문을 수행한다.
- P: -p와 같지만 패키지파일이름을 표준입력에서 읽어들인다.

정보선택

- -i: 패키지의 이름, 판본, 설명 등의 정보를 출력한다. 만약 —queryformat가 주어져있다면 그것을 리용하여 출력한다.
- R: 현재 패키지가 의존하고있는 패키지목록을 보여준다. (—requires과 같음)
- provides: 패키지가 제공하는 기능들을 보여준다.
- l: 패키지안의 파일을 보여준다.
- s: 패키지안의 파일상태를 보여준다. (-l포함) 매 파일의 상태는 normal(정상), not installed(설치되지 않음), replaced(교체)의 값을 가진다.
- d: 문서파일만 보여준다.(-l포함)
- c: 설정파일만 보여준다.(-l포함)
- scripts: 설치 및 제거과정에 사용되는 쉘스크립트가 있다면 그 내용을 출력한다.
- dump: 경로, 크기, 수정날자, MD5검사합, 방식, 소유자, 그룹, 설정파일정보, 문서파일정보, rdev 및 기호련결정보와 같은 파일정보들을 추출한다. -l, -c, -d 추가선택들중에서 최소한 하나가 사용되여야 한다.

검증추가선택

rpm 검증추가선택의 일반적인 형태는 다음과 같다.

```
rpm -V|-y|—verify [검증추가선택]
```

설치되여있는 파일들에 대하여 rpm자료기지에 저장된 내용과 원본패키지의 내용을 비교한다. 검증내용은 크기, MD5검사합, 호출권한, 형식, 소유자, 그룹 등이다. 차이점이 나타나면 그에 대하여 출력한다. 패키지지시추가선택은 패키지질문추가선택과 같으며 출력형식은 8개의 문자렬이다. "c"는 설정파일을 의미한다. 매 8개 문자는 RPM 자료기지에 저장된 속성과 비교한 결과를 나타낸다. "."은 이상이 없다는것을 나타낸다. 비교결과 문제점이 발견되면 다음과 같은 문자가 나타난다.

5	MD5검사합
S	파일크기
L	기호련결

T	갱신날자
D	장치
U	사용자
G	그룹
M	호출권한과 파일형식을 포함한 방식

서명확인

rpm서명확인지령은 다음과 같다.

```
rpm —checksig <패키지파일>+
```

패키지의 원본정보를 알아내기 위하여 패키지안에 있는 PGP서명을 검사한다. PGP 설정정보는 /etc/rpmrc에서 읽어들인다.

제거추가선택

rpm제거지령의 일반적인 형태는 다음과 같다.

```
rpm -e <패키지파일>+
```

- noscripts: preunistall, postuninstall스크립트를 실행하지 않는다.
- nodeps: 패키지를 제거하는동안 의존성을 검사하지 않는다.
- test: 패키지를 제거하지 않고 검사해본다

제작추가선택

rpm제작지령의 일반적형식은 다음과 같다.

```
rpm -bO [제작추가선택] <패키지 spec>+
```

-bO는 제작단계와 제작할 패키지를 나타내는것으로서 다음의 값을 가질수 있다:
- bp: spec파일의 "%prep"단계를 실행한다. 보통 원천을 풀고 수정하는 작업이다.
- bl: "목록검사"를 한다. "%files"부분은 확장마크로이다. 이 파일들이 존재하는가를 알아본다.
- bc: "%build"단계를 수행한다.(prep단계를 거친 후) 보통 make에 해당하는 작업을 수행한다.
- bi: "%install"단계를 수행한다.(prep, build단계를 거친 후) 보통 make install에 해당하는 일을 한다.

- bb: 2진패키지를 만든다.(prep, build, install단계를 수행한 후)
- ba: 2진 및 원천패키지를 만든다.(prep, build, install단계를 수행한 후)

다음의 추가선택도 함께 사용할수 있다.

- short-circuit: 중간단계를 거치지 않고 지정한 단계로 직접 이동한다. -bc와 -bi와만 함께 쓸수 있다.
- timecheck: "시간검사"를 0(불가능)으로 설정한다. 이 값은 rpmrc에서 "timecheck:"로 설정할수 있다. 시간검사값은 초로 표시되는데 파일을 묶는데 걸리는 최대시간을 정한다. 시간을 초과하는 파일들에 대하여 경고통보문이 출력된다.
- clean: 패키지를 만든 후 build등록부를 지운다.
- test: 어떠한 build단계도 거치지 않는다. spec파일을 검사할 때 유용하다.
- sign: 패키지안에 PGP서명을 써넣는다. 패키지를 누가 만들었는가를 확인할수 있다.
- rpm --recompile <원천패키지파일>+ 또는 rpm —rebuild <원천패키지파일>+ : rpm은 주어진 원천패키지를 설치하고 예비작업, 콤파일, 설치를 해준다. --rebuild는 새로운 2진패키지도 만들어준다. 제작을 마치면 build등록부는 --clean추가선택에서와 마찬가지로 지워진다. 패키지로부터 풀어졌던 원천과 spec파일은 삭제된다.

기존의 RPM에 서명하기

```
rpm —resign <2진패키지파일>+
```

패키지파일에 새로운 서명을 하며 이때 원래의 서명은 삭제된다.

서명기능을 사용하기 위해서는 PGP를 사용할수 있어야 한다. 그리고 RPM 공개열쇠를 포함하는 공개열쇠고리를 찾을수 있어야 한다. 기본적으로 RPM은 PGPPATH에서 지시하는 PGP기본설정을 사용한다. PGP가 기본적으로 사용하는 열쇠고리를 가지고있지 않을 때는 /etc/rpmrc파일에 설정해두어야 한다.

pgp_path는 /usr/lib/rpm대신 쓰이는 경로이다. 이 경로에는 자기의 열쇠고리가 포함되여있어야 한다. 자체로 만든 패키지에 서명을 하려면 자기의 공개열쇠와 비밀열쇠 한쌍을 만들어 두어야 한다.(PGP지도서 참고) /etc/rpmrc에 설정되여있는것외에 다음의 사항들을 추가해야 한다.

- signature: 서명 류형. 현재로서는 pgp만 지원된다.
- pgp-narne: 자체의 패키지에 서명할 사용자이름을 적는다. 패키지 제작시 —sign추가선택을 추가한다. 입력을 받으면 패키지가 만들어지고 동시에 서명된다.

자료기지재생성 추가선택

rpm 자료기지를 다시 만드는 지령은 다음과 같다.

```
rpm —rebuilddb
```

패키지에서 풀어진 원천과 spec파일은 삭제된다.

rpm은 강력한 패키지관리도구로서 매 소프트웨어패키지를 만들고 설치하고 질문하고 검증하고 갱신하며 제거할수 있다. 패키지란 설치할 파일들과 이름, 판본, 설명을 포함하는 패키지정보가 보관된 파일이다. 기본적인 7가지 동작방식(설치, 질문, 검증, 서명확인, 제거, 제작 그리고 자료기지 재구축)이 있으며 서로 다른 추가선택들을 가진다.

1) RPM 파일 읽는 법

패키지이름은 일정한 규칙으로 되여있다. 실례로 다음과 같은 Apache RPM 패키지가 있다고 가정한다.

apache-1.3.26-i686.rpm

- apache : RPM패키지이름이다.
- 1.3.26 : linux패키지의 판본이다.
- i686: 실행2진코드의 체계구성방식이다.

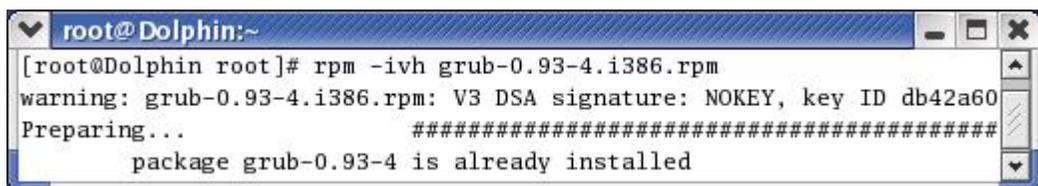
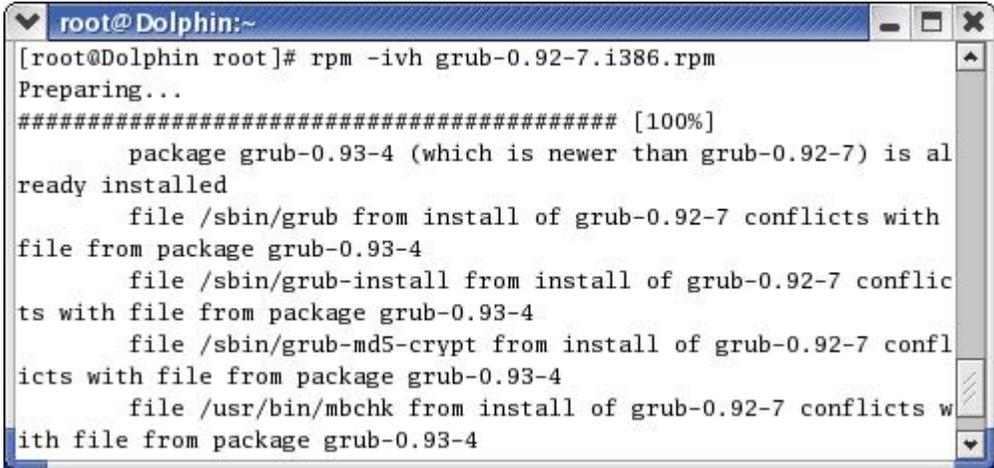
2) RPM 패키지설치

보통 RPM을 설치하려면 rpm -ivh지령뒤에 설치하려고 하는 패키지이름을 지정하면 된다.

```
# rpm -ivh apache-1.3.22-i686.rpm
```

하지만 우의 지령으로 rpm이 설치되지 않는 경우가 있을수 있다. 즉 설치시 기존파일이나 의존성관계 등으로 여러가지 오유가 나타날수 있다. 오유해결방법에 대하여 크게 세가지로 알아보자.

- 이미 패키지가 설치되여있는 경우

이러한 오유에 대한 해결방도는 —replacepkgs추가선택을 추가적으로 입력하여 덧쓰기하는것이다.

- 설치될 패키지의 파일이 충돌하는 경우

```
[root@Dolphin root]# rpm -ivh grub-0.92-7.i386.rpm
Preparing...
########################################### [100%]
        package grub-0.93-4 (which is newer than grub-0.92-7) is already installed
        file /sbin/grub from install of grub-0.92-7 conflicts with file from package grub-0.93-4
        file /sbin/grub-install from install of grub-0.92-7 conflicts with file from package grub-0.93-4
        file /sbin/grub-md5-crypt from install of grub-0.92-7 conflicts with file from package grub-0.93-4
        file /usr/bin/mbchk from install of grub-0.92-7 conflicts with file from package grub-0.93-4
```

일반적으로 패키지에 포함된 파일의 서고가 있는데 현재 설치하는 패키지가 같은 서고의 다른 판본을 설치하려고 하는 경우 정확하게 원인을 해결해야 한다. 이것을 해결하려면 --replacepkgs추가선택을 사용할수 있다.

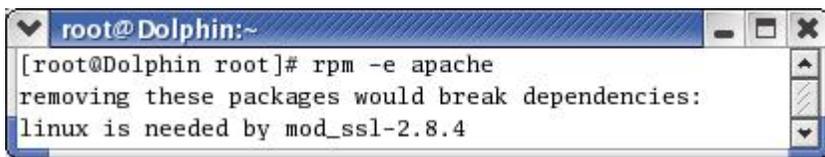
- 의존성문제가 발생하는 경우

이 경우에는 패키지를 설치하기 전에 먼저 설치해야할 패키지가 있다는것을 의미하므로 오유로 나타나는 패키지를 먼저 설치하면 된다.

3) 패키지 제거

rpm 패키지제거는 rpm -e추가선택과 함께 제거하려는 패키지이름을 지정하면 된다. 패키지제거는 패키지이름만 입력하여도 가능하다.

```
# rpm -e apache
```

이것도 역시 의존성문제로 오유통보문을 나타낼수 있다.

```
[root@Dolphin root]# rpm -e apache
removing these packages would break dependencies:
linux is needed by mod_ssl-2.8.4
```

제거하려는 패키지가 의존하고 있는 패키지를 먼저 제거하여야 한다. 의존성을 무시하고 강제로 제거하려면 --nodeps추가선택을 사용할수도 있지만 주의해야 한다.

4) 패키지 갱신

현재의 패키지를 새로운 패키지로 갱신할 때에는 rpm -Uvh추가선택을 사용한다.

만일 보다 낮은 판본으로 교체할 필요가 있으면 --oldpackage추가선택을 사용한다.

5) 패키지정보확인

rpm -ql추가선택뒤에 패키지이름을 지정하면 지정한 패키지가 설치된 등록부와 파일들을 볼수 있다.

설치된 파일이나 등록부로 해당 패키지 찾기

설치된 파일이 어느 패키지에 포함되여있는지 알아야 할 경우가 많다. 이런 경우에는 rpm -qf추가선택과 함께 찾으려고 하는 파일을 절대경로로 지정하면 지정한 파일에 해당하는 패키지이름을 보여주게 된다.

rpm2cpio

/usr/bin/rpm2cpio

RPM변환

지령형식: rpm2cpio [파일이름]

rpm2cpio는 rpm파일이름을 지적하면 cpio파일을 표준출력으로 보여준다. 만약 파일이름을 지적하지 않으면 표준입력으로부터 rpm파일을 받아들인다.

```
# rpm2cpio rpm-1.1-1.i386.rpm

# rpm2cpio < glint-1.0-1.i386.rpm
```

원격쉘의 사용

지령형식: rsh [추가선택] 콤퓨터이름 [지령]

-l username : 콤퓨터에 접근할 사용자이름을 지정한다.

원격콤퓨터의 쉘을 사용하여 지령을 실행할수 있게 하는 지령이다.

rsh지령은 원격쉘(remote shell)로서 원격콤퓨터에서 지령을 실행한다. rsh도 보안상 결함을 가지고있으므로 ssh지령을 사용하는것이 좋다.

rsh지령은 지적된 원격콤퓨터에 접속한 후 사용자가 제공한 지령을 수행하도록 한다. 만일 원격콤퓨터 hana에 대하여 ls를 실행하려면 다음과 같이 실행한다.

```
# rsh hana "ls -al /"
```

Red Hat 9.X판본에서는 보안상 문제로 하여 /etc/inetd.conf설정파일이 /etc/xinetd.d/rlogin으로 변경되였다. 이 파일을 살펴보면 기정값으로 rsh를 허용하지 않는다.

```
root@Dolphin:~
[root@Dolphin root]# cat /etc/xinetd.d/rsh
# default: on
# description: The rshd server is the server for the rcmd(3) routine a
#       consequently, for the rsh(1) program.  The server provides \
#       remote execution facilities with authentication based on \
#       privileged port numbers from trusted hosts.
service shell
{
        socket_type             = stream
        wait                    = no
        user                    = root
        log_on_success          += USERID
        log_on_failure          += USERID
        server                  = /usr/sbin/in.rshd
        disable                 = yes
}
```

r지령의 위험성

r지령 다시말해서 rcp, rlogin, rsh 등과 같이 원격콤퓨터에서 지령들을 수행하기 위한 지령들은 BSD(Berkeley Software Distribution)계렬에서 제공한다. 이것은 원격콤퓨터의 쉘이나 지령을 망을 통해 실행하는것으로서 사용자의 접근권한 등을 인증하는 과정을 포함하고있다. 하지만 원격으로 편리하게 작업할수 있는 반면에 인증절차를 콤퓨터와 계정만으로 통과하는 단순성으로 하여 보안상 문제가 있을수 있다는 결함도 있다. 더우기 위험한것은 "+"문자를 사용하여 임의의 사용자가 접근하는것도 허용할수 있다는 점이다.

체계관리자가 자리를 뜬 기회를 리용하여 관리자권한으로 .rhosts파일에 "++"와 같은 내용을 써넣은 다음 rlogin으로 root권한을 얻어 공격하는 사건이 자주 있다. 따라서 대부분의 체계에서는 r지령을 안전한 s지령(ssh, scp 등)으로 교체하여 사용하고있다.

rstat

/usr/bin/rstat

체계정보제공

허용된 콤퓨터에 체계정보를 보여주는 지령이다.

rstat는 cpu와 가상기억기 사용상태, 망가동시간, 하드디스크에 대한 정보를 제공하는 기능을 한다. rstat도 역시 보안상 결함이 있다.

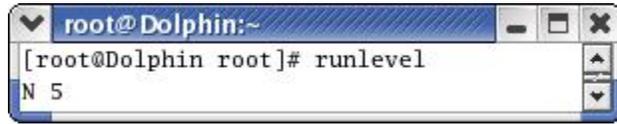
runlevel

/sbin/runlevel

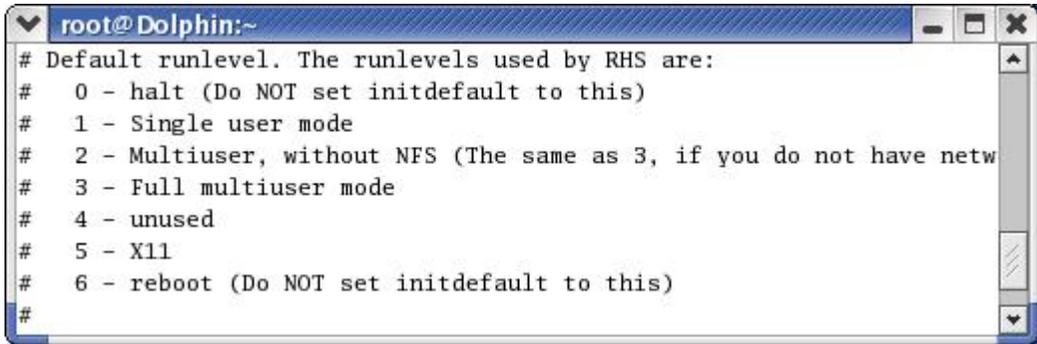
실행준위보기

지령형식: runlevel

이전체계의 실행준위와 현재체계의 실행준위을 공백문자로 나누어 표준출력으로 보여주는 지령이다. 만약 이전체계의 실행준위가 없으면 N을 표시한다. runlevel 지령은 System-V의 who -r지령을 대신해서 rc스크립트안에서 사용될수 있다. 또한 init

의 새로운 판본에서 이 정보들은 RUNLEVEL과 PREVLEVEL환경변수를 지정하는데 사용될수 있다.

```
[root@Dolphin root]# runlevel
N 5
```

이 체계는 현재 실행준위는 5로서 XWindows가 기동하도록 설정되여있다.

/etc/inittab파일에서 매 실행준위에 대한 설명을 주석으로 보여준다. Linux 배포판마다 조금씩 다를수 있지만 /etc/inittab파일을 조사하면 매 준위가 어떤것인지 확인할수 있다.

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have netw
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
```

rwall

/usr/bin/rwall

통보문보내기

지령형식: rwall 콤퓨터이름 [파일]

원격사용자가 망에 있는 모든 사용자에게 통보문을 보내는 지령이다. 이것은 보안상 매우 위험하다. Red Hat 7.X 이전 판본은 /etc/inetd.conf에서 봉사를 막도록 하였으며 7.X판본부터는 rwalld데몬을 실행하지 않도록 한다.

데몬을 중지한 후 ntsysv편의프로그람으로 기동할 때 자동실행되는 rwalld데몬을 제거한다.

```
[root@Dolphin root]# /etc/rc.d/init.d/rwalld stop
Stopping rwalld:                                          [FAILED]
```

rwho

/usr/bin/rwho

접속자확인

지령형식: rwho [추가선택]

who지령의 망판본으로서 현재 등록가입된 사용자의 정보를 보여주는 지령이다. 허용된 콤퓨터들에 who지령의 결과와 같은 정보를 보여주게 된다.

역시 보안상 결함이 있다. Red Hat 7.X 이전 판본들에서는 /etc/inetd.conf에서 봉사를 진행하지 않도록 설정한다. 7.X판본부터는 rwhod데몬을 실행하지 않도록 한다. 데몬을 중지한 다음 ntsysv편의프로그람으로 기동할 때 자동실행되는 데몬에서 제거한다.

script

/usr/bin/script

본문포착

지령형식: script [-a] [파일]

-a : 새로 기록하는것이 아니라 이전의 파일에 덧붙인다.

말단기에서 발생하는 모든 작업내용을 기록하는 지령이다. 이전 작업내용을 알아보는데 편리하다.

script지령만을 실행할 경우에는 기정으로 typescript파일에 말단기에서 진행된 작업들의 내용이 보관된다. 만일 script지령에 파일이름을 지정하면 지정한 파일에 작업내용

이 저장된다. -a추가선택은 지정한 파일이나 파일을 지정하지 않을 경우 기정보관파일인 typescript파일에 추가적으로 말단기의 작업내용을 보관한다.

실행중에 있는 script지령을 완료하기 위해서는 exit나 [Ctrl+D], logout지령을 입력한다.

다음의 실례는 말단에서 일어나는 작업들을 typescript라는 파일에 보관한다.

```
root@Dolphin:~
[root@Dolphin root]# script
Script started, file is typescript
[root@Dolphin root]# rm *.rpm
rm: remove regular file `grub-0.92-7.i386.rpm'? y
rm: remove regular file `grub-0.93-4.i386.rpm'? y
[root@Dolphin root]# mkdir dir
[root@Dolphin root]# ls
1                 install.log         newgrouptest2
1~                install.log.syslog  test1
1.dup             ispelltest          test2
2.dump            jointest            text
anaconda-ks.cfg   jointest2           text.txt
Desktop           lastlog             text.txt~
dir               linux.bt            typescript
d.txt             ls.txt              upgrade
d.txt~            ls.txt~             upgrade.tar.gz
firewall_role     newgrouptest1       Work
[root@Dolphin root]# exit
Script done, file is typescript
```

진행된 작업내용을 보기 위해서는 typescript파일을 본다.

```
# cat typescript
```

scp

/usr/bin/scp

원격파일복사

지령형식: scp [추가선택] [[user@host:] file1 ...]

-p : 원본파일의 변경시간, 호출시간, 방식을 변경하지 않고 그대로 보존한다.
-r : 지정한 등록부의 부분등록부 및 파일까지 복사한다.
-v : 진행상태를 구체적으로 보여준다.
-F ssh_config : 지정한 설정파일을 사용한다.
-P port : 지정한 포구로 접속을 시도한다.

원격콤퓨터의 파일을 복사하는 보안복사지령으로서 신원인증 및 암호화를 써서 telnet, rlogin, rcp, rsh를 안전하게 해준다.

다음의 실례는 webmaster계정으로 linuxroot.co.kp콤퓨터의 /home/webmaster/ 등록부에 있는 모든 파일을 접근하려는 콤퓨터의 작업등록부로 복사한다.

```
# scp webmaster@linuxroot.co.kp:/home/webmaster/* ./
```

지정한 등록부의 부분등록부까지 모두 복사하려면 -r추가선택을 사용한다.

아래의 실례는 admin계정의 권한으로 linuxroot.co.kp콤퓨터의 /home/admin/경로에 있는 모든 부분등록부와 파일을 국부콤퓨터의 현재등록부안에 있는 backup 등록부에 복사한다.

```
# scp admin@linuxroot.co.kp:/home/admin/* ./backup
```

setfdprm

/usr/bin/setfdprm

플로피디스크설정

 지령형식: setfdprm [추가선택] 장치이름 [이름]

-c 장치이름 : /etc/fdprm파일의 환경설정을 지운다.
-n 장치이름 : 초기화검색통보문을 내보내지 않는다.
-p 장치이름 [이름] : 지정한 이름으로 장치의 설정을 바꾼다.
-y 장치이름 : 초기화검색통보문을 내보낸다.

setfdpm지령은 플로피장치를 자동검색할수 있도록 디스크파라메터를 지정하거나 이전 설정을 지우고 검사통보문을 활성/비활성화한다.

fdformat /dev/fd0H1440지령으로 플로피디스크를 초기화한다.

setfdprm지령으로 플로피디스크이름이 1440/1440인 장치를 /dev/fd0장치이름으로 지정하여보자.

```
# setfdprm /dev/fd0 1440/1440
```

우의 실례와 달리 새롭게 지정한 장치이름으로 형식화할수 있다.

```
# fdformat /dev/fd0
```

setup

/usr/sbin/setup

체계설정

지령형식: setup

이 지령은 체계관련설정을 차림표방식으로 설정하는 편의프로그람이다. 사용자정보설정, 방화벽설정(ipchains), 건반설정, 마우스설정, 망설정, 인쇄기설정, 기동시 데몬선택, 음성기판설정, 시간대조절, XWindows설정 등을 할수 있다.

차림표에 대한 설명은 다음과 같다.

차림표이름	설 명	실행지령
Authentication configuration	사용자정보설정	authconfig
Firewal configuration	방화벽(ipchains)설정	lokkit
Keyboard configuration	건반설정	kddconfig
Mouse configuration	마우스설정	mouseconfig
Network configuration	망설정	netconfig
Printer corfiiguration	인쇄기설정	printconf-tui
System service	기동시 데몬설정	ntstysv

Sound card configuration	음성기판설정	sunconfig
Timezone configuration	시간대설정	timeconfig
X configuration	XWindows설정	Xconfigurator

setup지령으로 이 차림표전체를 볼수도 있으며 실행지령을 입력하여 매 차림표를 설정할수도 있다.

sh

/bin/sh

기본쉘의 호출

기본쉘을 호출하는 지령이다.

본래 /bin/sh는 기본쉘을 의미했지만 지금은 /bin/bash를 련결하여 사용한다. 다음과 같이 확인해볼수 있다.

만일 쉘스크립트로 작성한 ftpdown_count.sh라는 파일이 있다면 ./지령이나 sh지령으로 실행할수 있다.

```
$ sh ftpdown_count.sh
```

자세한 설명은 쉘스크립트부분을 참고하기 바란다.

shar

/usr/bin/shar

쉘압축편의프로그람

뉴스그룹 등에서 사용하며 쉽게 내용을 열람할수 있는 압축편의프로그람이다.

shar로 압축된 파일은 #!/bin/sh머리부와 포함파일을 풀기 위한 지령들로 구성된 쉘스크립트로서 압축되지 않은 파일들이 묶여있는 형태이다. 이 파일은 뉴스그룹관리자가 원

천파일을 쉽게 읽고 정리하거나 삭제할수 있으므로 지금도 Internet새소식그룹에서 종종 쓰이고있다. 대체로는 tar/gzip를 리용한다.

shar지령으로 압축된 포함파일은 unshar지령으로 해제할수 있다.

shar지령으로 압축하기

c원천파일을 shar로 압축해보자.

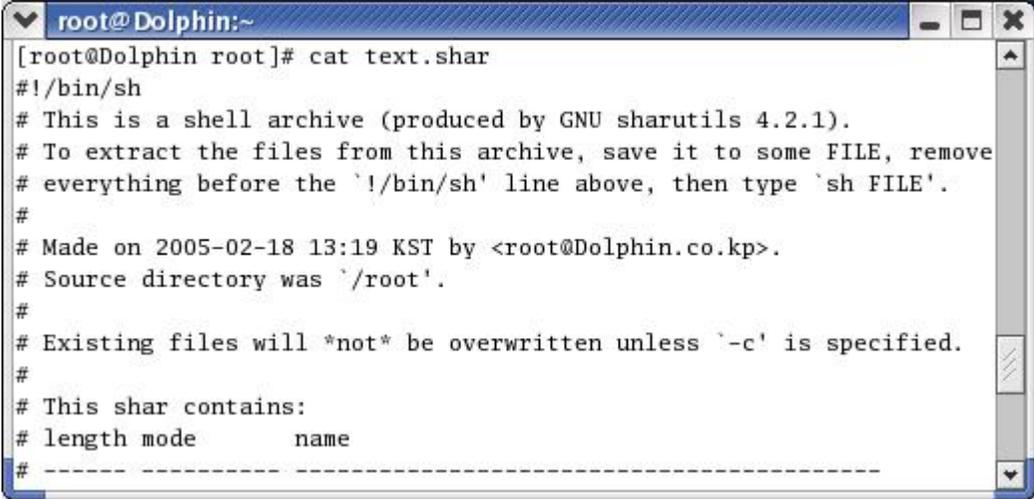
```
$ shar -o[저장할 파일] [원본파일]
```

또는 다음과 같은 방법을 사용할수도 있다.

```
$ shar [원본파일]>[저장할 파일]
```

그러면 hello.c원천파일을 hello.shar파일로 압축해보자.

```
$ sh hello.c>hello.shar
```

shar형태로 된 파일을 살펴보면 쉘스크립트형식으로 되여있다.

```
root@Dolphin:~
[root@Dolphin root]# cat text.shar
#!/bin/sh
# This is a shell archive (produced by GNU sharutils 4.2.1).
# To extract the files from this archive, save it to some FILE, remove
# everything before the `!/bin/sh' line above, then type `sh FILE'.
#
# Made on 2005-02-18 13:19 KST by <root@Dolphin.co.kp>.
# Source directory was `/root'.
#
# Existing files will *not* be overwritten unless `-c' is specified.
#
# This shar contains:
# length mode       name
# ------ ---------- ------------------------------------------
```

shar지령으로 압축 풀기

압축된 hello.shar파일을 해제하기 위해서는 unshar지령을 사용한다.

```
$ unshar hello.shar
```

련관지령

unshar .shar 파일압축해제지령

포함파일(archive)

포함파일은 여러 파일을 하나로 묶어놓은것으로서 대체로 여벌복사 혹은 여러개 파일들을 한번에 이동시키기 위해 사용한다. 포함파일은 단순히 파일들의 목록만을 포함할수도 있지만 하나의 등록부나 파일들의 구조일람표형식으로 구성될수도 있다. UNIX/Linux에서는 많은 경우 이런 포함파일을 작성하고 해체하는데 편의프로그람 tar를 리용한다.

showmount

/usr/bin/showmount

탑재정보 보기

지령형식: showmount [추가선택] [콤퓨터이름]

-a, --all : "콤퓨터이름:등록부"형식으로 모든 원격탑재정보를 보여준다. 콤퓨터이름은 의뢰기의 이름이고 등록부는 탑재된 파일체계의 등록부정보이다.
-d, --directories : 의뢰기에서 원격으로 탑재한 등록부목록을 보여준다.
-c, --exports : 반출된 등록부의 목록을 보여준다.
-h, --help : 도움말을 보여준다.
-v, --version : 판본정보를 보여준다.

showmount지령은 NFS/NIS지령으로 지정한 체계에서 원격으로 탑재한 의뢰기 목록들을 보여준다. 콤퓨터를 지정하지 않으면 현재 봉사기에서 탑재된 등록부의 정보를 보여준다.

점대점(point-to-point)

점대점이란 두 장치사이에 중개장치가 없이 두 장치를 직접 련결한 방식을 말한다.
이 방식은 콤퓨터들사이에 계속적으로 대화를 나누며 빠른 응답을 요구하는 경우와 콤퓨터체계가 다른 대형콤퓨터에 련결되여 말단처럼 사용되는 경우에 주로 리용된다.

shutdown

/sbin/shutdown

체계완료

지령형식: shutdown [추가선택] 시간 [경고통보문]

-c : 진행중인 shutdown지령을 취소한다.
-f : 재기동할 때 fsck지령을 건너뛰고 빠른 재기동을 한다.
-h : shutdown지령이 끝난 후 체계를 완료한다.
-k : 경고통보문만을 보여주고 실제로 체계는 끄지 않는다.
-n : 체계를 끄기 위한 init를 호출하지 않고 shutdown을 진행한다.
-r : 체계를 완료한 다음 재기동한다.
-t sec : 지정한 시간에 체계를 재기동한다. 지정할수 있는 값들은 다음과 같다.
now : 지금 당장 완료한다.
+m : 지정한 m분 이후에 완료한다.
hh:mm : 몇시(hh) 몇분(mm)에 완료한다.

정해진 지시사항에 따라 체계의 프로쎄스를 완료하고 기억기의 내용을 저장한 다음 안전하게 완료하는 지령이다.

shutdown지령은 체계를 안전하게 완료시키는 지령이다. 현재 접속중인 모든 사용자는 체계가 완료된다는 통보문을 받고 등록가입작업을 완료한다.

체계를 즉시 완료하기 위해서는 다음과 같이 실행할수 있다. 이것은 halt, init 0지령과 류사한 기능이다. 체계가 완료된 다음에는 재기동한다. 이 지령은 reboot, init 6 지령과 같다.

```
# shutdown -h now
```

참고로 일정한 시간이 지난 후에 전원끄기를 취소하려면 shutdown -c추가선택을 사용한다.

```
# shutdown -c now
```

slattach

/sbin/slattach

직렬회선련결

지령형식: slattach [추가선택] [tty]

-c command : 접속이 끊어졌을 때 실행하는 지령을 규정한다.
-d : 오유수정방식
-h : 접속이 끊어졌을 때 탈퇴한다.
-q : 어떤 통보문도 보여주지 않는다.
-l : /var/lock에 UUCP형의 잠금파일을 생성한다.
-n : mesg n지령과 동일하다.
-m : 8bit방식으로 초기화하지 않는다.
-e : 초기화한 후 즉시 빠져나온다.
-L : 3선 작용을 활성화한다.
-p proto : 규약종류를 지정한다. 여기에는 slip, adaptive, ppp, kiss 등이 있다.
-s speed : 회선을 련결하는 속도를 지정한다.

직렬회선을 망대면부로 련결하는 지령이다.

slattach지령은 TTY회선을 망대면부에 지정하는 역할을 하며 점대점접속으로 다른 콤퓨터를 리용할수 있게 한다.

SLIP망대면부를 직접 tty1포구에 련결할수 있다.

```
# slattach /dev/tty1
```

sleep

/bin/sleep

대기

 지령형식: sleep [추가선택] 수자 [단위]

--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

쉘에서 사용하는 wait와 같다. 지정한 시간(s)동안 아무 일도 하지 않고 대기상태로 된다.
지정가능한 수값의 단위에서 기정값은 초이다. 아래와 같이 지정한 수자와 단위로 시간을 지정할수 있다.

s(초), m(분), h(시간), d(일)

slocate

/usr/bin/slocate

파일찾기

 지령형식: slocate [추가선택]

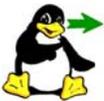
-u : /경로에서부터 slocate자료기지를 만든다.
-U dir : 지정한 등록부경로에서부터 slocate자료기지를 만든다.
-c : -u나 -U추가선택을 사용하는 경우 /etc/updatedb.conf의 위치를 추적한다.
-e dir1, dir2, ... : -u나 -U추가선택을 사용하는 경우 지정한 등록부들을 제외한다.
-f fs_type1, ... : -u나 -U추가선택을 사용하는 경우 지정한 파일체계를 제외한다.
-l level : 보안준위로서 0이면 보안검사를 하지 않으며 대신 검색이 좀 더 빠르다. 기정값은 1로서 보안검사를 진행한다.
-q : 오유통보문을 보여주지 않는다.
-n num : 지정한 수만큼의 파일만을 보여준다.
-i : 검색할 때 사용자에게 일일히 문의한다.
-r regexp : POSIX표현방식으로 자료기지를 검색한다.
-o file : 지정한 파일이름으로 자료기지파일을 만든다.
-d path : 지정한 경로에서 자료기지를 찾는다.
-h, --help : 사용법을 보여준다.
-v. --verbose : 자료기지를 생성할 때 구체적인 통보문을 보여준다.
-V, --version : 판본정보를 보여준다.

자체 자료기지를 통하여 파일을 찾아주는 지령으로서 locate지령의 보안을 강화한 판본이다.

체계가 자동적으로 slocate자료기지를 갱신하지 않으면 파일이나 등록부정보를 제대로 확인할수 없다. 때문에 체계에서는 updatedb프로그람을 하루에 한번씩 cron으로 실행하여 자료기지를 갱신한다.

Red Hat기준으로 하루에 한번씩 실행하는 cron설정등록부인 /etc/cron.daily/slocate.cron파일을 살펴보면 /usr/bin/updatedb파일이 실행되는것을 볼수 있다.

주의 최근의 배포판에서는 대체로 locate지령에 slocate를 련결하고있으므로 locate지령을 사용하여도 같은 결과를 얻을수 있을것이다.

find지령과 달리 slocate지령은 자료기지를 미리 생성하므로 파일을 빨리 찾을수 있게 한다. 실례로 httpd.conf파일을 slocate지령으로 찾아보자.

련관지령

find	조건에 맞는 파일을 찾아 지정한 지령을 수행하는 지령
locate	자체 자료기지를 통하여 파일을 찾아주는 지령

sort

/bin/sort

내용정렬

지령형식: sort [추가선택] [파일]

정렬추가선택

-b, --ignore-leading-blanks : 공백을 무시한다.
-d, --dictionary-order : 공백과 자모들만 비교한다.
-f, --ignore-case : 모든 문자를 소문자로 인식한다.
-g, --general-numeric-sort : 수자값을 비교정렬한다.
-i, --ignore-nonprinting : 인쇄가능한 문자만 비교한다.
-M, --month-sort : 날자를 비교하여 월순서로 정렬한다.
-n, --numeric-sort : 문자형수값으로 비교하여 정렬한다.
-r, --reverse : 비교결과를 반대로 보여준다.

기타 추가선택

-c, --check : 입력된 파일이 정렬되였는가를 검사한다.
-k, --key=pos1[, pos2] : 열쇠를 pos1에서 시작하여 pos2까지 지정한다.
-m, —merge : 여러개의 입력파일들을 합친다.
-o, --output=file : 지정한 파일에 결과를 보관한다.
-S, --buffer-size=size : 지정한 크기로 주기억완충기를 사용한다.
-t, --field-separator=sep : 마당구분자로 지정한다.
-T, --temporary-directory=dir : 림시등록부로 $TMPDIR나 /tmp를 사용하지 않고 지정한 dir를 사용한다.
-u, --unique : 마당안에서 같은 값은 제거하고 유일한 값만을 결과값으로 보여준다.
--help : 도움말을 보여준다.
--version: 판본정보를 보여준다.

sort지령은 지정한 파일로부터 입력을 받아 내용을 자모순서로 정렬하고 그 결과를 표준출력으로 보여준다. 만일 입력파일이 2개이상이라면 파일들을 합치여 하나의 파일로 정렬한다.

다음의 실례는 /etc등록부의 내용을 정렬하여 보여준다.

만일 마당안에서 중복되는 행을 보고싶지 않을 때에는 -u추가선택을 사용한다. 이것은 uniq지령과 같은 역할을 한다.

만일 파일에서 대소문자의 구별을 없애려면 -f추가선택을 사용한다. -f와 -u추가선택을 함께 리용한 실례를 아래에 보여주었다.

source

내부지령

환경설정 반영하기

지령형식: source 파일

스크립트나 환경설정파일을 읽어들여 환경변수를 변경한 다음 즉시 반영할수 있게 하는 지령이다. 만일 아래와 같이 Oracle을 설치하기 위한 환경변수를 $HOME/bash_profile에 추가했다고 하자.

```
root@Dolphin:~
[root@Dolphin root]# cat $HOME/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
```

다음과 같이 입력하면 변경된 환경변수를 직접 읽어들일수 있다.

```
$ source $HOME/.bash_profile
```

/usr/bin/split

파일분할

지령형식 : split [추가선택] [INPUT[PREFIX]]

-b, --bytes=SIZE : 지정한 크기만큼 파일을 만든다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

split지령은 파일을 작은 쪼각으로 나누는 편의프로그람으로서 플로피디스크에 여벌복사하거나 전자우편에 첨부하기 위하여 파일을 나눌 때 리용할수 있다. 아래에서는 현재등록부에 있는 모든 파일을 5MB크기로 압축하여 split.tar.gz이름을 가진 파일을 만드는 실례를 보여준다. 5MB가 넘으면 다른 이름의 파일로 분할압축된다.

```
[root@Dolphin img]# tar cvzf - *|split -b 5m - split.tar.gz
0001.jpg
0002.jpg
0003.jpg
0004.jpg
.....
0039.jpg
0040.jpg
0041.jpg
```

압축파일의 용량이 5MB를 넘어가면 split.tar.gz파일의 이름뒤에는 aa가 붙고 다시 split.tar.gzab로 련이어 압축한다. 마찬가지로 다른 파일들도 ac, ad, … 등과 같은 이름으로 만들어진다.

```
[root@Dolphin img]# ls
0001.jpg  0010.jpg  0019.jpg  0028.jpg  0037.jpg
0002.jpg  0011.jpg  0020.jpg  0029.jpg  0038.jpg
0003.jpg  0012.jpg  0021.jpg  0030.jpg  0039.jpg
0004.jpg  0013.jpg  0022.jpg  0031.jpg  0040.jpg
0005.jpg  0014.jpg  0023.jpg  0032.jpg  0041.jpg
0006.jpg  0015.jpg  0024.jpg  0033.jpg  split.tar.gzaa
0007.jpg  0016.jpg  0025.jpg  0034.jpg  split.tar.gzab
0008.jpg  0017.jpg  0026.jpg  0035.jpg  split.tar.gzac
0009.jpg  0018.jpg  0027.jpg  0036.jpg  split.tar.gzad
```

이렇게 분리된 파일들을 하나로 합치기 위해서는 cat지령을 사용한다.

```
# cat split.tar.gza* >> split.tar.gz
```

주의 >>는 기존파일의 내용뒤에 새로운 파일을 추가할 때 사용한다.

KDE (K Desktop Environment)

KDE는 UNIX작업기용 원천공개형도형방식의 탁상환경이다. KDE는 처음에 "Kool Desktop Environment"이라고 불렀으며 Internet상의 공식적인 KDE우편목록과 많은 새소식그룹 그리고 IRC통로 등에서 광범한 토론을 통하여 현재도 계속 개발하고있는 프로젝트이다. KDE는 완전한 GUI와 함께 파일관리자, 창문관리자, 도움말체계, 구성관리체계, 도구 및 편의프로그람 그리고 여러가지 응용프로그람들을 포함하고있다. KDE응용프로그람들중에서 가장 유명한것은 KOffice이다.

KDE는 현재 Linux, Solaris, FreeBSD, OpenBSD 및 LinuxPPC 등에서 사용되고있으며 세계의 많은 프로그람작성자들이 KDE개발에 참가하고있다.

SSH의뢰기

지령형식 : ssh [추가선택] 콤퓨터이름 [지령]

-l user : 지정한 사용자이름으로 등록가입한다.
-n : /dev/null파일로부터 입력을 받아들인다.
-F config : 지정한 파일을 설정파일로 한다.(기정은 ~/.ssh/config)
-A : 인증대리인처리를 능동으로 한다.
-a : 인증대리인처리를 비능동으로 한다.
-I 파일 : 공개열쇠인증과 같은가 확인한다.(기정은 ~/.ssh/identity)
-v : 오유수정통보문을 구체적으로 현시한다.
-q : 아무런 경고통보문도 현시하지 않는다.
-e char : 탈퇴문자렬을 설정한다.
-c chiper : 암호화알고리듬을 선택한다.
-m macs: 규약판본2에 대한 MAC알고리듬을 규정한다.
-p port : 지정한 포구로 접속한다. 봉사기는 같은 포구를 사용하여야 한다.
-N : 쉘이나 지령을 수행하지 않는다.
-g : 원격콤퓨터들이 포구에 접속할수 있게 한다.
-o 'option' : 지정한 추가선택을 설정파일에서 읽어들인것처럼 처리한다.
-b addr : 국부IP주소를 보여준다.

암호화를 진행하여 안전한 접속을 제공하는 원격SSH봉사기접속용 의뢰기프로그람이다. SSH는 Secure SHell의 략자이다. 이전의 Telnet는 봉사기와의 련결상태에서 파케트송수신이 모두 망에 로출되기때문에 불법침입자들이 내용을 훔쳐볼수 있었다. 이와 같은 문제점들은 ssh를 사용하면 해결할수 있다.

```
$ ssh -l admin linuxroot.co.kp
```

또는 다음과 같이 입력할수도 있다.

```
$ ssh admin @ linuxroot.co.kp
```

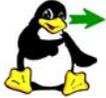
SSH봉사기에 처음 접속하면 아래와 같은 통보문이 현시된다.

```
The authenticity of host 'linuxroot.co.kp (192.168.8.85)' can't be established.
RSA key fingerprint is 39:ff:a3:70:bd:20: 16:a6:7c:5e:d4:7a:60:37:0c:f7.
```

```
Are you sure you want to continue connecting (yes/no)?
```

이제 yes를 입력하면 RSA열쇠를 자기의 국부체계의 $HOME/.ssh/known_hosts 파일에 보관한다.

그후에는 원격말단과 같이 접속하여 사용하면 된다.

련관지령

scp　　ssh봉사기에서 원격으로 복사할수 있는 지령

stat　　/usr/bin/stat

상태보기

지령형식 : stat [추가선택] 파일이름

-l : 련결파일의 원천파일정보를 보여준다.
-i : 파일이름이 있는 파일체계의 정보를 보여준다.
-v : 판본정보를 보여준다.
-t : 요약된 정보를 보여준다.

stat지령은 파일이나 파일체계의 상태를 보여주는 지령으로서 파일이나 파일체계의 크기, 블로크, IO블로크, 호출날자, 수정날자 등을 볼수 있게 한다.

아래와 같이 test1파일을 한번 살펴보자. 파일의 이름, 크기, 블로크, IO블로크, 장치 및 inode와 같은 정보들을 자세히 보여준다.

-f추가선택은 지정한 파일이 있는 파일체계의 정보를 보여준다.

```
[root@Dolphin root]# stat -f test1
  File: "test1"
    ID: 0          Namelen: 255     Type: ext2/ext3
Blocks: Total: 1512435    Free: 130263     Available: 53435
      Size: 4096
Inodes: Total: 768544     Free: 474232
```

등록부를 지정하여 등록부의 상태를 볼수도 있다.

```
[root@Dolphin root]# stat /
  File: `/'
  Size: 4096            Blocks: 8           IO Block: 4096
Directory
Device: 306h/774d       Inode: 2            Links: 23
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)   Gid: (
 0/    root)
Access: 2005-02-18 13:32:08.000000000 +0900
Modify: 2005-02-18 12:05:12.000000000 +0900
Change: 2005-02-18 12:05:12.000000000 +0900
```

Linux에서 만들수 있는 파일의 최대크기

이것은 사용하는 파일체계에 따라 다르다. 이전에 사용하던 minix나 ext파일체계에서는 2GB, ext2파일체계는 64GB 크기까지 만들수 있었다. 그것은 ext2의 inode는 12개의 블로크와 직접 련결되여 한개는 간접, 한개는 2중, 한개는 3중으로 련결되여있으며 ext2 zone은 256개까지의 zone지시기를 사용할수 있기때문이다.

ext3인 경우 4TB크기까지 파일을 만들수 있으며 Reiser나 XFS 등은 그이상의 크기를 가진 파일도 만들수 있다.

strace

프로그람에서 실행하는 프로세스 추적

 지령형식 : strace [추가선택] 지령 [매개변수]

strace지령은 체계호출과 신호를 추적하여 진단하고 오유수정해주는 도구로서 프로그람에서 실행되는 프로세스가 어떤것인가를 추적하는 지령이다. 이외에도 2진파일에 어떤 경로이름이 콤파일되여있는가를 보거나 조건을 살펴보는 경우 혹은 2진파일의 동작원리를 알 필요가 있을 경우에도 사용할수 있다.

아래의 실례를 보자. 이 실례에서는 ls지령에 대한 여러가지 콤파일경로나 조건, 동작에 대한 정보를 볼수 있다.

```
root@Dolphin:~
[root@Dolphin root]# strace ls
execve("/bin/ls", ["ls"], [/* 37 vars */]) = 0
uname({sys="Linux", node="Dolphin.co.kp", ...}) = 0
brk(0)                                  = 0x80586c8
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_A
NONYMOUS, -1, 0) = 0x40016000
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such
 file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=115232, ...}) = 0
```

strings

/usr/bin/strings

문자렬찾기

 지령형식 : strings [추가선택] 파일

-, -a, --all : 목적파일전체를 검색한다.
-f, --print-file-name : 매 문자렬앞에 파일이름을 보여준다.
-min-len, -n min-len, --bytes=min-len : 최소문자렬의 길이를 지정한다.(기정값은 4)
-o : -t추가선택과 비슷하다.
-t base, --radix=base : 지정한 형식으로 매 문자렬앞에 편위값을 보여준다. 다음은 지정할수 있는 base형식들이다
  d : 편위값을 10진수로 기록한다.
  o : 편위값을 8진수로 기록한다.
  x : 편위값을 16진수로 기록한다.
--target=format : 2진파일의 형식을 규정한다.
-v, --version : 판본정보를 보여준다.

목적파일(.obj) 또는 2진파일에서 정상적으로 출력할수 있는 문자렬을 찾는 지령이다. 기정문자렬은 4문자이다.

먼저 hello.c원천파일을 cat지령으로 보자. cat지령은 원천파일을 있는 그대로 모두 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# cat hello.c
#include <stdio.h>
main()
{
        printf("test");
}
```

hello.c파일을 strings지령으로 다시 한번 살펴보자. 결과 아래와 같이 4문자이상인 문자렬만 검색하여 보여준다.

```
root@Dolphin:~
[root@Dolphin root]# strings hello.c
#include <stdio.h>
main()
        printf("test");
```

지정한 문자렬을 기준으로 문자렬을 찾을수 있다. 17문자이상인 문자렬만을 찾아보자. 최소문자렬을 지정하기 위해서는 -17 혹은 -n 17 혹은 —bytes=17로 지정할수 있다. 결과 한 행의 길이가 18자인것만 현시된다.

```
[root@Dolphin root]# strings -17 hello.c

#include <stdio.h>
```

baud

baud는 자료전송속도를 나타내는 단위로서 회선을 통하여 1s동안 전달되는 자료 bit수를 말한다. bps단위와 함께 모뎀전송속도를 표기하는 단위로 많이 쓰인다. bps는 초당 전송되는 bit수로서 만일 세개의 bit가 한 신호단위를 이룬다면 baud속도는 bps속도의 1/3이 된다.

strip

/usr/bin/strip

.obj파일용량 축소

지령형식 : strip [추가선택] 목적파일이름

-F bfdname, --target=bfdname : 입력파일을 지정한 파일로 한다.
-O format, --output-target=format : 파일을 지정한 형식으로 출력한다.
-R section, --remove-section=section : 지정한 절(section)을 삭제한다.
-s, --strip-all : 모든 기호들을 제거한다.
-S, -g, --strip-debug : 오유수정기호만 제거한다.
--strip-unneeded : 위치를 재지정하는 정보를 제외한 모든 기호들을 지운다.
-K symbol, --keep-symbol=symbol : 원천파일에서 지정한 기호이름은 삭제하지 않는다.
-o filename : 지정한 .obj파일을 정리하여 다른 파일로 저장한다.
-x , --discard-all : 일반기호를 제외한 나머지 기호들을 지운다.
-X, --discard-locals : 콤파일러의 위치정보를 지운다.
-V, --verbose : 진행과정을 자세히 보여준다.

strip지령은 .obj파일에서 기호들을 제거하여 파일의 용량을 줄인다. 여러가지 추가선택을 리용하면 기호들을 선택적으로 제거할수 있다. strip지령으로는 회선번호정보, 위치재지정정보, debug부분, typchk부분, 주석부분, 파일머리부(header) 등 모든 기호 또는 그 일부를 제거할수 있다. 기호가 제거된 .obj파일은 실제실행에는 영향을 주지 않는다.

file지령으로 현재 파일이 정리되였는가를 알수 있다. 파일정보 마지막에 "stripped"이라고 되여있으면 정리된 파일을 나타내며 "not stripped"로 되면 정리되지 않은 파일을 나타낸다.

libmenu.so.5.3파일을 정리하여 용량을 줄여 보자. -v는 정리한 결과를 보여주고 -o는 정리된 파일을 다른 파일로 저장한다.

결과 2.3MB정도의 파일용량이 1.8M정도로 줄어들었다.

Object코드

CPU가 직접 실행할수 있는 기계어코드이다. 고급언어로 작성된 프로그람은 콤파일러 등을 통해 기계어로 변환된다.

stty

/bin/stty

말단기의 환경설정

지령형식 : stty [설정...]
stty [추가선택]

조종설정

[-]parenb : 출력에서 기우성비트를 생성하고 입력에서 기우성비트를 요구한다.
[-]parodd : 기우성설정("-"는 우수).
cs5 cs6 cs7 cs8 : 문자의 크기를 5, 6, 7, 8bit로 설정한다.
[-]hupcl [-]hup : 마지막 프로쎄스가 tty를 닫으면 Hangup신호를 보낸다.
[-]cstopb : 문자당 두개의 정지비트를 사용한다. ("-"는 하나)
[-]cread : 입력을 받아들인다.
[-]clocal : 모뎀조종신호를 비능동상태로 설정한다.
[-]crtscts (np) : RTS/CTS로 주고받기를 진행한다.

입력설정

[-]Ignbrk : 중단지령을 무시한다.
[-]brkint : 중단지령을 주었을 때 새치기신호를 발생시킨다.
[-]ignpar : 기우성오유를 무시한다.
[-]parmrk : 기우성오유를 표시한다.
[-]inpck : 입력에 대한 기우성검사를 진행한다.
[-]istrip : 입력문자의 상위(8번째)비트를 지운다.
[-]inlc : 행바꾸기문자를 되돌이문자로 바꾼다.
[-]igncr : 되돌이문자를 무시한다.
[-]icrnl : 되돌이문자를 행바꾸기문자로 바꾼다.
[-]ixon : XON/XOFF흐름조종을 설정한다.
[-]ixoff [-]tandem : 체계의 입력완충기억기가 거의 차면 정지문자를 보내도록 한다. 그 다음

완충기억기가 비면 시작문자를 보낸다.
[-]iuclc (np) : 대문자를 소문자로 바꾼다.
[-]ixany (np) : 어떤 문자든 다시 출력할수 있게 한다. ("-"는 시작문자만 허용)
[-]imaxbel (np) : 완충기억기가 찬 상태에서 문자가 도착하면 경고음을 내고 입력완충기억기는 갱신하지 않는다.

출력설정
[-]opost : 후처리출력.
[-]olcuc (np) : 소문자를 대문자로 바꾼다.
[-]ocrnl (np) : 되돌이문자를 행바꾸기문자로 바꾼다.
[-]onlcr (np) : 개행문자를 되돌이-행바꾸기문자로 바꾼다.
[-]onocr (np) : 첫번째 렬에서 되돌이문자를 출력하지 않는다.
[-]onlret (np) : 개행문자가 되돌이문자의 역할을 수행한다.
[-]ofill (np) : 지연시간대신 채움문자를 사용한다.
[-]ofdel (np) : NULL문자대신 Delete문자를 채움문자로 사용한다.
nl1 nl0 (np) : 새행의 지연격식.
cr3 cr2 cr1 cr0(np) : 되돌이문자의 지연격식.
tab3 tab2 tab1 tab0 (np) : 수평 Tab지연격식.
bs1 bs0 (np) : backspace의 지연격식.
vt1 vt0 (np) : Tab 지연격식.
ff1 ff0 (np) : Formfeed지연격식.

국부설정
[-]isig : 새치기, 완료, 중단 특수문자를 사용할수 있게 한다.
[-]icanon : erase, kill, werase, rprnt특수문자를 사용할수 있게 한다.
[-]iexten : POSIX특수문자가 아닌 문자를 사용할수 있게 한다.
[-]echo : 입력문자를 반사(echo)한다.
[-]echok : kill문자뒤에 행바꾸기문자를 반사한다.
[-]echonl : 다른 문자는 반사하지 않더라도 행바꾸기문자는 반사한다.
[-]xcase (np) : icanon이 설정되여있을 때 입출력시에 대문자를 해당 문자의 앞에 "\"를 붙여서 사용할수 있게 해준다.
[-]tostop (np) : 말단기에 쓰는것을 시도하는 배경작업을 멈추게 한다.
[-]echoprt [-]prterase (np) : "\"와 "/"사이에서 지워진 문자를 다시 반사한다.
[-]echoctl [-]ctlecho (np) : 조종문자를 글자그대로가 아니라 모자표기법 "^c"로 반사한다.
[-]echoke [-]crtkill (np) : echoctl과 echok설정대신 echoprt echoe설정값을 사용하여 한 줄의 매 문자를 지움으로써 kill특수문자를 반사한다.

조합설정
[-]evenp [-]parity : parenb pafodd cs7과 같다. "-"를 쓰면 -parenb cs8과 같다.
[-]oddp : parenb parodd cs7과 같다. "-"를 쓰면 parenb cs8과 같다.
[-]nl : -icrnl -onlcr와 같다. "-"를 쓰면 icrnl -inlcr -igncr onlcr -ocrnl-onlret와 같다.
ek : erase, kill특수문자를 원래의 값으로 되돌린다.

sane : cread -ignbrk brkint -inlcr -igncr icrnl -ixoff -iuclc ixany imaxbel opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0 isig icanon iexten echo echoe echok -echonl noflsh -xcase -tostop -echoprt echoctl echoke와 같으며 모든 특수문자를 원래의 값으로 되돌린다.
[-]cooked : brkint ignpar istrip icml ixon opost isig icanon과 같으며 min, time문자가 같으면 eof, eol문자 또한 원래의 값으로 되돌린다. "-"를 사용하면 raw와 같다.
[-]raw : -ignbrk -brkint -ignpar -parrnrk inpck -istrip -inlcr -igncr -icrnl -ixon -ixoff -iuclc -ixany -imaxbel -opost -isig -icanon -xcase min 1 time 0과 같다. "-"를 사용하면 cooked와 같다.
[-]cbreak : -icanon와 같다.
[-]pass8 : -parenb -istrip cs8와 같다. "-"를 사용하면 parenb istrip cs7과 같다.
[-]litout : -parenb -istrip -opost cs8과 같다. "-"를 사용하면 parenb istrip opost cs7과 같다.
[-]decctlq (np) : -ixany와 같다.
[-]tabs (np) : tab0과 같다. "-"를 사용하면 tab3과 같다.
[-]case [-]LCASE (np) : xcase iuclc olcuc와 같다.
crt : echoe echoctl echoke와 같다.
dec : echoe echoctl echoke -ixany와 같으며 또한 새치기문자를 Ctrl+C, erase를 Del, kill을 Ctrl+U로 설정한다.

특수문자

특수문자의 기정값은 체계마다 다르다. "이름 값"과 같은 형식으로 설정한다. 여기서 이름은 아래와 같고 값은 글자그대로의 모자표시법("^c") 또는 16진수를 의미하는 "0x", 8진수를 나타내는 "0" 혹은 10진수로 표기할수도 있다. 값에다 "^-"를 붙이는 경우 그 특수문자는 사용할수 없게 된다.

intr	새치기신호를 보낸다.
quit	완료신호를 보낸다.
erase	마지막 문자를 지운다.
kill	현재행을 지운다.
eof	파일의 끝을 알린다.(입력완료)
eol	한 행의 끝.
eol2 (np)	행끝을 나타내는 문자.
swtch (np)	다른 쉘계층으로 절환.
start	멈춰진 출력을 다시 시작한다.
stop	출력을 멈춘다.
susp	말단정지신호를 보낸다.
dsusp (np)	입력을 갱신한 후 말단정지신호를 보낸다.
rprnt (np)	현재행을 다시 그린다.
werase (np)	마지막 단어를 지운다.
lnext (np)	특수문자라 할지라도 다음 문자는 글자그대로 입력한다.

말단기의 행설정을 확인하고 수정하는 지령이다.

stty지령은 아무런 인수도 주어지지 않으면 전송속도, 회선조종번호, 변화된 회선설정값들을 보여준다.

어떤 기능앞에 "-"표시를 한것은 그 기능을 해제할수 있다는 뜻이다. 일부 인수들은 체계에 따라 사용할수 없는 경우도 있다.

su

/bin/su

다른 계정의 환경 사용하기

지령형식 : su [추가선택] [사용자] [쉘 변수]

-c COMMAND, --command=COMMAND : 대화형쉘을 시작하지 않고 -c추가선택을 쉘에 주어서 한개의 지령만을 수행하도록 한다.
-f, --fast : csh와 tcsh에만 해당되는것으로서 시작파일(.cshrc)을 읽지 않도록 한다.
--help : 사용법을 보여준다.
-, -l, --login : 사용자의 등록가입스크립트파일을 읽어 사용자의 환경으로 변경한다.
-m, -p, --pleserve-environment : "HOME", "USER", "LOGNAME", "SHELL" 등의 환경변수값을 변경하지 않는다.
-s, --shell shell : su를 실행하는 사용자가 관리자가 아니거나 사용자의 쉘이 제한되였으면 /etc/passwd에 등록된 사용자의 쉘을 수행하지 않고 지정한 쉘을 수행한다.
--version : 판본정보를 보여준다.

등록탈퇴하지 않고도 림시로 다른 사용자의 UID, GID를 사용할수 있도록 한다. 주로 관리자의 쉘을 사용할 때 많이 사용된다.

su는 어떤 사용자가 잠시 다른 사용자가 될수 있도록 해준다. 실제 사용자ID, 그룹ID 그리고 지정한 사용자의 보충적인 그룹으로 쉘을 실행한다. 사용자를 지정하지 않으면 기정으로 관리자인 root가 설정된다. 실행되는 쉘은 사용자의 통과어목록에서 찾으며 만일 없으면 /bin/sh를 수행한다. 사용자통과어가 설정되여 있다면 su는 실제 사용자ID가 0(관리자)이 아닌 이상 통과어를 문의한다.

su는 현재등록부는 변경하지 않는다. 사용자의 통과어항목으로부터 HOME, SHELL과 같은 환경변수를 설정하며 만일 관리자가 아니라면 USER와 LOGNAME을 지정한 사용자로 설정한다. 이 쉘은 등록가입쉘이 아니다.

실례를 통하여 한번 살펴보자. 아래와 같이 id지령으로 확인한 결과 현재 등록가입한 쉘이 admin이라는것을 알수 있다.

su지령으로 관리자권한쉘로 변경할수 있다.

```
$ su
```

참고로 su지령은 root의 환경변수들은 적용하지 못하므로 su -지령을 사용하는것이 좋다.

```
$ su -
```

또는 "$ su -l", "$ su --login"지령을 주어도 같은 결과를 얻을수 있다.

sum

/usr/bin/sum

파일검사확인

지령형식 : sum [추가선택] 파일

-r : 추가선택을 무효로 하고 BSD검사합알고리듬을 리용한다. 1KB블로크를 리용한다.
-s, --sysv : sys V검사합알고리듬을 리용하며 512byte블로크를 리용한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

sum지령은 지정된 파일을 읽어들여 해당 파일의 1KB단위의 블로크수와 검사합을 계산하는 역할을 하며 만일 추가선택을 지정하지 않으면 기정값인 BSD검사합알고리듬을 사용하게 된다. 이것은 보통 통신상에서 주고 받은 파일들사이에 이것이 원본파일에 대한 사본파일인가 또한 도중에 변경되지 않았는가를 알아보는데 리용할수 있다.

아래와 같이 원본파일과 대상파일을 지정하여 1024B당 블로크수와 검사합을 확인할 수 있다. 결과 이 파일은 원본파일과 동일하다는것을 알수 있다.

swapoff

/sbin/swapoff

교환기능중지

지령형식 : swapoff -a [장치이름]

-a : /etc/fstab에서 교환장치로 지정된 모든 장치의 교환기능을 해제한다.

파일 및 장치들에 설정했던 교환기능을 완료하는 지령이다.

련관지령

mkswap 교환령역을 지정하는 지령
swapon 교환기능을 설정하는 지령

교환(swap)령역

교환령역이란 주기억장치의 자료를 림시로 저장할수 있는 하드디스크의 공간으로서 조작체계에서 자료를 필요로 할 경우 주기억장치의 내용과 디스크상의 내용을 바꾸는데 리용한다. 교환기능은 체계의 주기억용량보다 큰 프로그람이나 파일을 다룰수 있게 한다.

swapon

/sbin/swapon

교환기능설정

지령형식 : swapon [추가선택] [장치이름]

-a : /etc/fstab에서 "swap"로 지정된 모든 장치를 유효하게 한다. 이미 swap로 실행되고 있는 장치는 그대로 넘어간다.
-p priority : swapon에서 사용될 우선권을 설정한다. 우선권은 0~32767사이의 값이다. pri=value라는 항목을 /etc/fstab의 추가선택마당에 써넣으면 swapon -a지령을 수행할 때 리용된다.

파일 혹은 경로로 분류되는 블로크장치를 교환매체로 설정하는 지령이다.

512MB의 교환파일 만들기

free지령으로 작성된 교환파일을 확인한다.

```
# free
```

교환기능을 해제하려면 swapoff지령을 사용하면 된다.

```
# swapoff/swap
```

sync

/bin/sync

기억기내용저장

지령형식 : sync [추가선택]

—help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

기억기의 내용을 디스크에 동기적으로 저장하는 지령이다.

핵심부는 속도가 뜬 디스크에 대한 읽기, 쓰기과정에 처리속도가 지연되는것을 피하기 위해 기억기에 자료를 저장한다. 이것은 실행속도를 높이지만 콤퓨터가 파손되면 자료가 손실되며 파일체계가 손상될수 있다. sync는 처리소자가 비정상적으로 정지되기전에 호출되여야 한다. 대체로 이 처리는 shutdown이나 reboot, halt지령으로 정지시킨다.

동기화(Synchronization)

동기화는 서로 다른 속도로 동작하는 장치들이 서로 원활하게 동작할수 있도록 동작의 진행을 일치시키는 작업으로서 UNIX에서는 기억기의 내용을 하드디스크에 저장하는것을 동기화한다고 말한다.

tac

/usr/bin/tac

파일내용 거꾸로 출력

지령형식 : tac [추가선택] 파일

-b, --before : 구분자를 앞에 붙여준다. 기정구분자는 행바꾸기문자이다.
-r, --regex : 정규표현식으로 구분자를 사용한다.
-s, --separator=string : 검색한 문자렬을 기준으로 문장을 자른다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일내용을 마지막행부터 보여주는 지령이다. 즉 cat지령이 보여주는 결과를 거꾸로 보여준다.

먼저 파일내용을 순서대로 출력하는 cat지령을 리용하여 파일을 살펴보자.

tac지령은 아래에서부터 웃방향으로 파일을 출력한다.

3번째 행을 기준으로 잘라서 그아래의 행을 먼저 출력하고 다시 그이전의 내용을 보여주려면 다음과 같이 "-s3"으로 "3"이라는 문자렬을 검색하여 잘라준다. 즉 3아래부분이 먼저 출력되고 다시 앞부분부터 3전까지 출력된다.

련관지령

cat　　파일을 우에서부터 아래방향으로 보여준다.
head　　파일의 첫 행부터 10행 또는 설정한 행만큼 보여준다.
tail　　파일의 마지막 행부터 10행 또는 설정한 행만큼 보여준다.
rev　　행마다 본문을 반대로 읽어들인다.

tail

/usr/bin/tail

파일의 뒤부분 보기

 지령형식 : tail ［추가선택］ 파일

-n[단위] : 지정된 단위에 따라 파일의 마지막부터 n번째까지 보여준다.(l 행, c 문자)
-c num[단위] --byte num[단위] : 마지막으로부터 지정한 수만한 byte를 보여준다.
　　b 512byte, k 1024byte
-q --quiet, --silent : 출력하기 전에 파일이름을 보여주지 않는다.(기정으로는 보여주지 않는다.)
-v, --verbose : 출력하기 전에 파일이름을 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

tail지령은 문서의 마지막행부터 시작하여 지정한 행까지 보여준다. 아무런 설정을 하지 않을 경우 10개 행을 출력한다. 파일의 앞부분을 출력하는 head지령의 반대로 생각하면 된다. 출력하려는 부분에 대한 설정에는 행을 사용하지만 설정에 따라 용량, 문자수로 구분하기도 한다.

보통 파일의 내용을 보여주는 지령으로는 cat, more, less 등이 있지만 이 지령은 많은 파일의 내용중에서 마지막 행만을 보고 싶을 때 리용한다. 보통 최근의 기록파일을 볼 때 많이 사용한다.

/var/log/messages기록파일을 tail지령으로 보도록 하자. tail지령은 파일의 마지막 10개 행을 보여준다. 알아보기 쉽게 "|"(흐름선)으로 nl지령을 입력하였다. 참고로 nl지령은 행수를 보여주는 지령이다.

```
[root@Dolphin root]# tail /var/log/messages | nl
     1  Feb 18 14:50:09 Dolphin gdm(pam_unix)[3076]: session closed fo
r user admin
     2  Feb 18 14:50:17 Dolphin gconfd (admin-4675): GConf server is n
ot in use, shutting down.
     3  Feb 18 14:50:17 Dolphin gconfd (admin-4675): Exiting
     4  Feb 18 14:50:21 Dolphin gdm(pam_unix)[3076]: session opened fo
r user root by (uid=0)
     5  Feb 18 14:50:21 Dolphin gconfd (root-4852): starting (version
2.2.0), pid 4852 user 'root'
     6  Feb 18 14:50:21 Dolphin gconfd (root-4852): Resolved address "
xml:readonly:/etc/gconf/gconf.xml.mandatory" to a read-only config sou
rce at position 0
     7  Feb 18 14:50:21 Dolphin gconfd (root-4852): Resolved address "
xml:readwrite:/root/.gconf" to a writable config source at position 1
     8  Feb 18 14:50:21 Dolphin gconfd (root-4852): Resolved address "
xml:readonly:/etc/gconf/gconf.xml.defaults" to a read-only config sour
ce at position 2
     9  Feb 18 14:50:24 Dolphin kernel: cdrom: This disc doesn't have
any tracks I recognize!
    10  Feb 18 14:55:26 Dolphin kernel: Adding Swap: 524280k swap-spac
e (priority -2)
```

-5추가선택은 마지막행부터 5개 행 즉 우에서 본 6번째부터 10번째 행이 출력된다.

```
[root@Dolphin root]# tail -5 /var/log/messages | nl
     1  Feb 18 14:50:21 Dolphin gconfd (root-4852): Resolved address "
xml:readonly:/etc/gconf/gconf.xml.mandatory" to a read-only config sou
rce at position 0
     2  Feb 18 14:50:21 Dolphin gconfd (root-4852): Resolved address "
xml:readwrite:/root/.gconf" to a writable config source at position 1
     3  Feb 18 14:50:21 Dolphin gconfd (root-4852): Resolved address "
xml:readonly:/etc/gconf/gconf.xml.defaults" to a read-only config sour
ce at position 2
     4  Feb 18 14:50:24 Dolphin kernel: cdrom: This disc doesn't have
any tracks I recognize!
     5  Feb 18 14:55:26 Dolphin kernel: Adding Swap: 524280k swap-spac
e (priority -2)
```

wc지령으로 messages기록파일의 행수는 6622라는것을 확인할수 있다.

```
[root@Dolphin root]# wc /var/log/messages
   6622   78176  509262 /var/log/messages
```

6622의 행중에서 6500행부터 나머지 마지막까지 보고싶은 경우 +6500추가선택을 사용한다.

```
# tail +6500 /var/log/messages
```

련관지령

cat 파일을 첫 행부터 아래방향으로 보여준다.
tac 파일을 마지막 행부터 웃방향으로 보여준다.
head 파일의 첫 행부터 10행 또는 설정한 행만큼 보여준다.

talk

/usr/bin/talk

1대1대화

지령형식 : talk [사용자ID] [tty이름]

사용자ID : talk로 접속할 계정
tty이름 : 같은 계정이라면 tty이름을 정해준다. tty** 또는 pts/X로 지정해준다. 현재 체계에 접속하고있는 계정은 "finger"지령으로 확인한다.

말단기에 접속한 사용자와 대화를 나눌수 있다.

talk는 말단기에서 사용하는 대화프로그람으로 생각하면 된다. 같은 체계에 있는 사용자나 다른 체계사용자와의 대화가 모두 가능하다. talk를 사용하기 위해서는 talk봉사기가 동작하고있어야 한다.

talk로 같은 체계에 접속한 다른 ID의 사용자와 접속하려면 "talk [사용자ID]"의 형식으로 사용한다. 만약 같은 ID로 접속한 다른 말단기의 사용자와 대화하려면 다음과 같이 tty이름을 붙여준다.

대화하려는 사용자말단기를 지정하면 상대방의 응답을 기다리게 된다.

상대편창문에는 다음의 통보문이 뜨면서 대화를 요청하고있다는것을 알려준다. 상대

방도 우에서 한것과 같이 접속해야 한다.

서로 talk로 접속을 하면 창문이 우아래 두개로 나누어져서 우에는 본인이 입력하는 내용이 현시되고 아래에는 상대방이 입력하는 내용이 실시간으로 현시된다.

현재 사용중에 있는 체계가 아닌 sec.linuxroot.com.kp체계에서 hana라는 ID와 대화하려고 한다면 다음과 같이 접속한다.

```
# talk hana@sec.linuxroot.com.kp
```

talk를 사용하려면 talk봉사기데몬이 시작되여있어야 하는데 일반적으로 Linux에서는 talk봉사기로서 ntalk를 사용한다.

ntalk (New Talk)

talk를 사용하려면 talk봉사기가 가동하고있어야 한다. talk는 Sun회사에서 개발한 OS에서 사용되던것으로서 최근 체계들에는 대체로 ntalk가 설치되여있다. ntalk는 Red Hat Linux체계에서 xinitd데몬에 포함되여있다. talk와 ntalk는 포구 517과 518에서 호환성이 없다.

IRC (Internet Relay Chat)

IRC는 일종의 대화체계로서 의뢰기/봉사기구조의 쏘프트웨어이다.

IRC사용자는 대화그룹(통로이라고도 부른다.)을 새로 만들거나 기존의 대화그룹에 참가할수 있다. 기존의 대화그룹과 이미 속해있던 성원들을 찾아보는 규약도 있다. 망의 종류에 따라 자기를 자칭하는 별명을 등록하여 대화중에도 계속 사용할수 있다. IRC규약은 TCP를 사용하며 대체로 6667번포구를 사용한다. IRC에서 다른 Linux사용자들을 만나려면 irc.hanirc.org로 접속하여 #linux나 #kldp 등의 대화그룹에 들어가면 된다.

tar

/bin/tar

파일묶기

 지령형식 : tar [추가선택] 파일이름1 파일이름2

파일이름1 : 압축이나 묶음으로 만들어진 결과파일의 이름을 지정한다.
파일이름2 : 압축이나 묶음으로 만들 원천파일들을 지정한다.
-x, --extract, --get : 포함파일을 푼다.
-c, --create : 새로운 포함파일을 만든다.
-v, --verbose : 처리되는 내용의 구체적인 정보를 보여준다.
-z, --gzip, --ungzip : gzip로 압축한다. ungzip로 압축을 해제한다.
-j, -I, --bzip : bzip2를 리용하여 압축한다.
-t, --list : 압축되는 목록을 보여준다.
-d, --diff, --compare : 압축파일과 파일체계사이의 차이점을 비교한다.
-f, --file=ARCHIVE : 기정으로 설정되는 파일이름을 지정할수 있다.
-r, --append : 포함파일의 끝에 새로운 파일을 추가한다.
-u, --update : 포함파일에 들어있지 않는 새로운 파일들만 추가한다.
-A, --catenate : 포함파일에 tar파일을 추가한다.
-w, --interactive : 모든 작업에 대한 확인을 요구한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

tar는 여러개의 파일이나 등록부를 하나의 파일로 묶는 역할을 한다. 주로 파일의 여벌복사나 파일전송을 위해 여러개의 파일들을 하나로 결합하는데 리용된다. 파일압축에는 gzip이나 bzip2가 많이 쓰인다.

파일묶기

html등록부를 backup.tar로 묶어보자. tar로 파일을 묶을 때 주로 -cf추가선택을 많이 쓴다. c는 새로 tar파일을 만드는 역할을 하며 f는 포함파일이름을 지정한다.

```
root@Dolphin:/var/www
[root@Dolphin www]# tar -cf backup.tar html
[root@Dolphin www]# ls -alh
total 292K
drwxr-xr-x    9 root      root         4.0K Feb 18 15:08 .
drwxr-xr-x   27 root      root         4.0K Feb 13 08:22 ..
-rw-r--r--    1 root      root         250K Feb 18 15:08 backup.tar
drwxr-xr-x    2 root      root         4.0K Feb 13 07:56 cgi-bin
drwxr-xr-x    3 root      root         4.0K Feb 13 07:21 error
drwxr-xr-x    4 webmaster webmaster      4.0K Feb 13 03:04 html
drwxr-xr-x    3 root      mailman      4.0K Feb 13 07:57 icons
drwxr-xr-x   13 root      root         4.0K Feb 13 07:21 manual
drwxr-xr-x    2 root      root         4.0K Feb 13 07:54 nut-cgi-bin
drwxr-xr-x    6 root      root         4.0K Feb 13 08:23 wordtrans
```

파일보기

-tvf추가선택을 쓰게 되면 tar로 묶인 파일을 풀지 않고 내용만 볼수 있다.

묶음풀기

-xf추가선택은 backup.tar파일을 푼다. 또한 -xvf추가선택을 사용하면 파일이 처리되는 내용을 구체적으로 볼수 있다.

```
#tar -xf backup.tar
```

묶음 압축하기

다시 html등록부내용을 묶음과 동시에 gzip로 압축할수 있다. -cf추가선택에 gzip의 z추가선택을 추가하면 압축까지 할수 있다. 파일이름은 tar파일과 구별을 위해 일반적으로 tar.gz나 tgz의 확장자를 사용한다.

```
root@Dolphin:/var/www
[root@Dolphin www]# tar -czf backup.tar html/
[root@Dolphin www]# ls -alh
total 116K
drwxr-xr-x    9 root      root          4.0K Feb 18 15:08 .
drwxr-xr-x   27 root      root          4.0K Feb 13 08:22 ..
-rw-r--r--    1 root      root           75K Feb 18 15:10 backup.tar
drwxr-xr-x    2 root      root          4.0K Feb 13 07:56 cgi-bin
drwxr-xr-x    3 root      root          4.0K Feb 13 07:21 error
drwxr-xr-x    4 webmaster webmaster     4.0K Feb 13 03:04 html
drwxr-xr-x    3 root      mailman       4.0K Feb 13 07:57 icons
drwxr-xr-x   13 root      root          4.0K Feb 13 07:21 manual
drwxr-xr-x    2 root      root          4.0K Feb 13 07:54 nut-cgi-bin
drwxr-xr-x    6 root      root          4.0K Feb 13 08:23 wordtrans
```

-xzf추가선택으로 backup.tar.gz파일을 압축해제할수 있다. 물론 gzip -d추가선택으로 gzip압축을 해제한 다음 tar -xf추가선택으로 tar묶음을 해제할수도 있다.

```
# tar -xzf backup.tar.gz
```

묶여있는 tar파일은 u추가선택을 통해 갱신할수 있다. 즉 원본파일이 변경되거나 파일목록을 추가할 때마다 tar파일을 새로 묶지 않고 변화된 파일만 갱신할수 있다. 첫번째 실례에서 만들어진 backup.tar를 갱신해보자.

image등록부가 backup.tar파일로 갱신된것을 보여준다.

tee

/usr/bin/tee

같은 내용을 두 경로로 동시에 출력하기

지령형식 : tee [추가선택] 파일

파일 : 보관될 파일이름
-a, --append : 덧쓰지 않고 표준입력한 내용을 주어진 파일에 첨부한다.
-i, --ignore-interrupts : 새치기신호를 무시한다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

tee지령은 입력한 내용을 화면에 보여주는 동시에 파일에 보관한다.

이미 있는 파일과 이름이 같은 경우에는 파일에 덧쓰기하며 "-a"추가선택을 사용하면 그 파일에 입력하는 내용이 첨부된다. 지령이나 흐름선을 통해 나오는 출력을 화면에 보여주거나 파일에 보관하지만 그 내용에는 아무런 영향도 미치지 않는다.

mailist의 내용을 흐름선으로 받아 sort지령으로 정렬하고 이 결과를 tee지령으로 화면에 출력하는 동시에 maillist_sort파일에 보관한다.

telinit

/sbin/telinit

프로쎄스조종

지령형식 : telinit runlevel

telinit지령은 현재 준위의 모든 프로쎄스를 완료하고 지정한 준위에 설정된 프로쎄스를 시작한다. /etc/inittab파일의 설정에 따라 준위를 수행한다.

Xwindows의 실행준위는 일반적으로 5이다. 이 상태에서 xdum을 모두 완료하려면 telinit를 리용하여 실행준위를 3으로 만들어준다.

```
# telinit 3
```

실행준위를 6으로 지정하면 체계는 완료된다.

```
# telinit 6
```

알파(Alpha)

알파는 DEC(Digital Equipment Corporation)회사에서 개발된 극소형처리기와 콤퓨터체계의 이름을 가리키는 용어이다. 알파처리기는 DEC회사의 주요제품인 VAX보다 새롭고 더욱 완성된 구성방식을 쓰고있다. 알파는 RISC방식기반의 처리기이며 한번에 64bit를 처리한다. DEC는 성능을 개선하기 위해 RISC방식을 추가하였다. 일반적으로 사용하는 Intel호환의 극소형처리기와는 다른 점이 많다.

telnet

원격접속

지령형식 : telnet [추가선택] [콤퓨터 [포구]]

콤퓨터 : 접속할 콤퓨터. Internet주소형식을 리용한다.
포구 : 접속에 리용할 콤퓨터의 포구이며 지정하지 않으면 telnet에서 사용되는 초기값으로 한다.(23번 포구)
-l UID : telnet봉사기체계에 접속할 계정을 사용자ID로 지정한다.
-a : 현재사용자이름을 ID로 하여 접속한다.

telnet지령을 리용하여 콤퓨터에 원격접속한다. 엄밀히 말하자면 원격체계를 지원하기 위한 지령이며 TCP/IP기반의 규약이다.

telnet는 접속하려는 봉사기의 IP 혹은 령역으로 접속할수 있다.

USER ID를 지정하면 접속시에 따로 등록가입하지 않아도 해당한 ID의 통과어입력으로 가입할수 있다. "-a"추가선택을 사용하면 현재 사용중에 있는 계정으로 가입한다.

```
# telnet -l hana www.linuxroot.co.kp
```

접속할 체계의 포구를 지정할수 있다. telnet봉사는 일반적으로 23포구를 사용한다. 그러나 봉사기의 보안이나 관리자의 특별한 리유로 하여 telnet의 봉사포구를 변경할 경우가 있다. 이런 경우 접속포구를 지정해준다.

봉사하는 포구에 직접 접속하여 봉사여부를 알수도 있다. 웨브봉사여부를 알아보자. Connected to통보문은 웨브봉사포구(80)가 열렸다는것을 알려준다.

아래는 FTP포구로 접속을 시도하였다. Connection refused통보문은 봉사가 거절되였다는것을 알려준다.

telnet사용지령

접속도중에도 [ctrl+]를 눌러 지령방식으로 들어간다.

지 령	설 명
logout	사용자가 등록탈퇴하며 접속을 끊는다.
display	인수를 출력한다.
mode	문장이나 문자의 방식을 변경한다. "mode ?"로 도움말을 볼수 있다.
open	콤퓨터나 IP를 적어 열고싶은 싸이트나 콤퓨터를 정한다.
quit	telnet를 완료한다.
send	특수문자를 보낸다. "send ?"로 도움말을 볼수 있다.
set	인수를 설정한다. "set ?"로 도움말을 볼수 있다.

unset	인수설정을 해제한다. "unset ?"로 도움말을 볼수 있다.
status	telnet상태를 출력한다.
toggle	인수를 고정한다. "toggle ?"로 도움말을 볼수 있다.
slc	특수문자의 상태를 변경한다. "slc ?"로 도움말을 볼수 있다.
z	telnet를 정지한다.
!	쉘지령을 사용할수 있다.
environ	환경변수를 변경한다. "environ ?"로 도움말을 볼수 있다.
?	도움말을 보여준다.

test

/usr/bin/test

파일검사

지령형식 : test [표현식]
test [추가선택]

--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

파일류형을 검사하고 값을 비교하는 지령이다.

조건문에서 사용되여 다음에 오는 파일 또는 문자렬 등의 인수를 검사한다. 참인 경우 0을 되돌리고 거짓인 경우 0이 아닌 값을 되돌린다. 조건문에서 편리하게 사용할수 있다.

실례를 들어보자.

간단한 if조건문이다. if문의 사용법을 모르는 경우 shell부분을 참고하기 바란다. 우의 실례에서 "-e"는 파일의 존재여부를 확인한다. 다시말하여 파일이 있으면 "true"값을 되돌리고 그렇지 않으면 그냥 완료된다. "true"를 출력하였으므로 "test"파일이 있다는것을 알수 있다.

확인하고싶은 파일이나 문자렬에 맞게 표현식을 사용하여 검사한다.

표현식	설명
-b	블로크특수파일이면 참
-c	문자장치파일이면 참
-d	등록부이면 참
-e	파일이면 참
-f	보통파일이면 참
-g	set-group-id파일이면 참
-k	점착비트설정파일이면 참
-L	기호련결파일이면 참
-p	흐름선파일이면 참
-r	읽기가능한 파일이면 참
-s	0보다 큰 크기를 가지는 파일이면 참
-S	파일이 존재하고 소케트파일이면 참
-t [fd]	fd가 말단기에서 열렸으면 참 (기정값은 1)
-u	파일이 존재하고 set-user-id비트가 설정되면 참
-w	쓰기가능파일이면 참
-x	실행가능파일이면 참
-O	파일이 존재하고 유효사용자ID의 소유이면 참
-G	파일이 존재하고 유효그룹ID의 소유이면 참
파일1 -nt 파일2	파일1이 파일2보다 최근에 생겼거나 수정되였다면 참
파일1 -ot 파일2	파일1이 파일2보다 오래된것이면 참
파일1 -ef 파일2	두 파일이 같은 장치, 같은 inode번호를 가지면 참
-Z	문자렬의 길이가 0이면 참
-n	문자렬의 길이가 0이 아니라면 참
문자렬1 = 문자렬2	두 문자렬이 같으면 참
문자렬1!= 문자렬2	두 문자렬이 같지 않으면 참
!표현식	표현식이 거짓이면 참
표현식1 -a 표현식2	표현식1과 표현식2가 둘 다 참이면 참
표현식1 -o 표현식2	표현식1 또는 표현식2 둘중에 하나라도 참이면 참

핵심부(kernel)와 Linux

핵심부는 조작체계의 핵심부분이라는 뜻이며 프로그람이 실행될 때 체계를 리용할수 있도록 중개하는 역할을 담당한다. 흔히 말하는 Linux는 핵심부와 체계프로그람 그리고 함께 패키지화된 일반 응용프로그람을 모두 포함해서 말하고있다. Linux를 GNU/Linux라고 부르기도 한다.

모든 조작체계에는 핵심부가 들어있지만 특히 Linux에 대해 말할 때 핵심부가 빠지지 않는것은 핵심부를 직접 콤파일하여 사용자에게 맞도록 설정하거나 자체의 변경정보를 추가할수도 있으리만큼 전용핵심부를 만들수 있기때문이다. 핵심부는 일반적으로 /등록부나 /boot등록부에 zImage라는 이름으로 압축되여 생성된다.

tftp

/usr/bin/tftp

tftp봉사기접속 의뢰기프로그람

지령형식 : tftp [콤퓨터]

TFTP는 FTP보다 간단한 기능의 TFTP(Trivial File Transfer Protocol)봉사를 하는 콤퓨터접속의뢰기프로그람이다. 사용자인증이 필요없으며 등록부구조를 보여주지 않는다.

일반사용법은 ftp와 같으며 원격콤퓨터접속은 IP나 령역이름을 지정하면 된다.

```
# tftp linuxroot.co.kp
```

tftp사용지령들은 다음과 같다.

지 령	설 명
?	사용법을 보여준다.
ascii	mode ASCII와 같다.
binary	mode binary와 같다.
status	현재의 tftp상태를 보여준다.

verbose	tftp상태를 구체적으로 보여준다.
mode	전송방식(ASCII 또는 binary)을 설정한다.
connect 콤퓨터 [포구]	지정한 콤퓨터와 포구로 접속을 시도한다.
get 파일이름	원격접속콤퓨터의 파일을 국부체계에 내리적재한다. 한번에 여러 파일을 지정하여 내리적재할수도 있다.
get 국부파일이름 원격파일이름	원격접속콤퓨터의 파일을 지정한 국부파일이름으로 내리적재한다.
put 파일이름	국부파일을 원격접속콤퓨터에 올리적재한다. 한번에 여러 파일을 지정하여 올리적재할수 있다.
put 국부파일이름 원격파일이름	국부파일이름을 원격접속콤퓨터에 지정한 원격파일이름으로 올리적재한다.
quit	tftp접속을 끝낸다.

사용자 public_html등록부를 자동으로 만들기

사용자계정에서 웨브봉사를 할 경우 home등록부의 public_html등록부에 index.html파일을 만들게 된다. 이것은 Apache설정파일인 httpd.conf의 UserDir 마당에서 지정한 등록부를 생성하게 한다. 기정으로 UserDir public_html이 설정되여있다.

사용자계정을 추가할 때마다 자동으로 public_html등록부를 만들게 하려면 /etc/skel등록부에 public_html등록부를 만들어주면 된다.

top

프로세스상태

지령형식: top [추가선택]

-b : 묶음방식으로 정보를 출력한다. 정보를 화면에 한행씩 실시간으로 현시한다.
-d delay : 지정한 시간(s)만큼 정보를 갱신하여 보여준다.
-i idle : 프로세스나 허울(zombie)프로세스를 보여주지 않는다.
-n num : 지정한 수자만큼 갱신정보를 보여주고 완료한다.
-p pid : 지정한 프로세스ID의 정보만을 보여준다.
-q : 시간지연이 없이 갱신된 정보를 계속 보여준다.
-s : 일부 대화식지령을 비능동으로 한다. (Secure방식)
-S : 루적자료를 보여준다.(cumulative방식)

top지령은 5s에 한번씩 체계프로세스상태를 갱신하여 보여준다. 현재시간, 체계갱신시간, 체계에 등록가입한 사용자수, 지난 1, 5, 15min간의 체계평균부하를 보여준다. 다음 행에는 프로세스정보, CPU상태, 기억기와 교환상태를 보여준다.

주의 갱신시간(uptime)은 체계의 동작시간을 의미한다.

top설정지령

지령	설명
space	정보를 갱신한다.
^L	화면을 다시 갱신한다.
F, f	마당을 추가하거나 제거한다. 아래 "top 항목"을 참조한다. 정보를 보려는 항목의 영어자모를 눌러 선택/취소한다.
O, o	표시되는 마당의 정렬순서를 바꾼다. 아래 "top 항목"을 참조한다. 순서를 설정한다. 대문자로 선택한 항목을 누르면 왼쪽으로 이동하고 소문자를 누르면 오른쪽으로 이동한다.
h, ?	top지령으로 정보를 보는 상태에서 실시간프로세스를 실행할 수 있는 지령을 보여준다.

k	프로쎄스를 끌낸다.
n, #	표시할 프로쎄스의 수를 지정한다.
s	표시하는 정보의 갱신시간을 지정한다.
w	~/.toprc에 설정된 내용을 보관한다.
q	top지령을 완료한다.

top보기수정

기본창문에서 상단의 정보를 수정하는데 쓰인다.

지령	설명
s	실시간정보를 루적방식으로 보여주거나 해제한다.
i	idle프로쎄스정보를 보여주거나 해제한다.
I	Irix나 Solaris정보를 보여주거나 해제한다.
c	지령행에서 실행된 지령자체를 보여주거나 해제한다.
l	평균적재정보를 보여주거나 해제한다.
m	기억기정보를 보여주거나 해제한다.
t	요약된 정보만을 보여주거나 해제한다.

top정렬

기본창문에서 실행하는 지령으로서 현재정보를 사용자의 요구대로 정렬한다.

지령	설명
r	프로쎄스의 우선순위를 변경한다.
N	pid정보로 정렬한다.
A	age정보로 정렬한다.
p	CPU사용량으로 정렬한다.
M	적재된 기억기사용량으로 정렬한다.
T	시간/루적시간으로 정렬한다.
u	지정한 사용자관련정보만을 보여준다.

top항목

마당의 구성을 변경할 때 나오는 내용을 정리하였다.

부호	기호	이름	설명
*	A	PID	프로쎄스ID
	B	PPID	부모프로쎄스ID
	C	UID	사용자ID
*	D	USER	사용자이름
*	E	%CPU	CPU사용량
*	F	%MEM	기억기사용량
	G	TTY	사용중에 있는 tty를 보여준다.
*	H	PRI	우선순위
*	I	NI	Nice Value
	J	PAGEIN	페지오유수(Page Fault Number)
	K	TSIZE	코드크기(kb)
	L	DSIZE	자료+탄창크기(kb)
*	M	SIZE	가상사본크기(kb)
	N	TRS	현재문자크기(kb)

	O	SWAP	교환크기(kb)
	P	SHARE	분할된 페지수(kb)
	Q	A	접근한 페지수(kb)
	R	WP	쓰기보호된 페지수(kb)
	S	D	버림페지수
*	T	RSS	현재 페지가 상주하고있는 크기
	U	WCHAN	휴식상태의 함수
*	V	STAT	프로쎄스상태
*	W	TIME	CPU시간
*	X	COMMAND	지령
	Y	LC	마지막 사용된 CPU
	Z	FLAGS	과제기발

touch

/bin/touch

빈 파일 만들기

 지령형식: touch [추가선택] 파일이름

-a, --time=atime, --time=access, --time=use : 최근 파일사용시간만 갱신한다.
-c, --no-create : 지정한 파일이 없더라도 파일을 생성하지 않는다.
-d time, --date time : 현재시간대신 지정한 시간으로 변경한다.(월이름, 시간대, am/pm)
-m, --time=mtime, --time=modify : 파일이 수정된 시간을 변경한다.
-r file, --reference file : 지정한 파일의 시간으로 변경한다.
-t time : 현재시간대신 지정한 시간으로 변경한다. [[cc]yy]mmddhhmm[.ss]) [cc]yy:년, mm:달, dd:일, hh:시, mm:분, ss:초
--help : 도움말을 보여준다.
--version: 판본정보를 보여준다.

touch지령은 주어진 파일의 최근사용시간과 최근변경시간(파일이 수정된 시간)을 체계의 현재시간으로 바꾼다. 만일 파일이 없다면 빈 파일을 만들게 된다.

이 지령은 파일에 대한 접근/수정시간을 현재 시간이나 특정한 시간으로 바꾸어주는 편의프로그람으로서 새 파일을 만들 때에도 리용할수 있다.

touch는 파일이 존재하는 경우 파일의 접근 및 수정시간을 변경한다. 시간을 지정하

지 않으면 체계의 현재시간이 적용된다.

touch -c추가선택을 사용하면 지정한 파일이 없는 경우 새 파일을 만들지 않는다. 또한 -t추가선택을 리용하면 새로 만드는 파일의 생성시간을 현재시간이 아닌 시간으로 설정할수 있다.

2005년 6월 6일 00시 00분으로 파일의 수정날자를 변경해보자.

YYMMDDHHMM(YY:년도의 뒤자리, MM:월, DD:일, HH:시간, MM:분)형식으로 입력한다.

tr

/usr/bin/tr

문자 변환 및 삭제

지령형식: tr [추가선택] 문자렬1 [문자렬2]

-c, --complement : 문자렬1을 ASCII값(001~337)과 비교하여 보수연산을 진행한다.
-d, --delete : 문자렬1에서 지정한 문자를 지우고 표시한다.
-s, --squeeze-repeats : 문자렬2에서 반복되는 문자를 지운다.
-t, --truncate-set1 : 문자렬1을 문자렬2의 길이로 자른다.
--help : 도움말을 보여준다.
--version: 판본정보를 보여준다.

문자렬에서 특정문자를 삭제하거나 변환한다.

tr는 문자변환려과기로서 인용부호나 []를 적절하게 사용하여 표준출력으로 내보내는 역할을 한다.

아래의 실례는 지정한 파일의 내용가운데서 모든 소문자를 대문자로 수정한것이다.

-d추가선택은 지정한 파일의 내용에서 지정한 문자렬을 모두 지우고 내용을 표시한다. 다음의 실례는 smallletter파일에 들어있는 모든 수자들을 지워버린것이다.

tr는 반복되는 문자를 정리할수도 있다. 다음의 실례는 -s추가선택에 " "(공백)을 지정해주어 련속된 공백을 삭제하고 삭제한 공백을 행바꾸기문자인 "\012"로 바꾸어준다. (ASCII코드 012는 행바꾸기문자(newlines)이다)

보통 등록부의 크기를 나타낼 경우 du지령을 사용한다. 여기서 -s추가선택을 사용하여 지정한 등록부의 부분등록부까지 포함한 총 사용량을 보여준다.

traceroute

/usr/sbin/traceroute

파케트이동경로 보기

 지령형식: traceroute [추가선택] 콤퓨터이름 [파케트크기]

-m : hop를 지정한다.(기정값은 30이며 최대로 255까지 지정할수 있다.)
-n : 주소찾기를 중지할수 있다.
-p : 시작포구번호를 지정한다.
-q : 집합당 파케트수를 지정한다.
-v : 구체적인 정보를 보여준다.
-w : 시간초과값을 지정한다.

지정한 콤퓨터로 가는 망경로를 보여주는 지령이다.

어떤 콤퓨터가 망에 련결되여있는가에 대한 검사에서 대표적인것이 ping지령이다. 하지만 ping은 단순히 련결성만을 나타내므로 콤퓨터에서 ICMP파케트를 받지 않게 설정하면 정확한 련결성검사결과를 알수 없다.

traceroute지령은 실제파케트가 지나가는 경로를 확인하여 콤퓨터에 련결된 망위험구간을 알아내고 망련결성도 검사할수 있는 지령이다.

현재 콤퓨터에서 주소가 192.168.8.84인 콤퓨터에 련결된 망경로를 볼수 있다.

true

/bin/true

참값반환

 지령형식: true [추가선택]

--help : 도움말을 보여준다.
--version: 판본정보를 보여준다.

항상 참값을 반환하는 지령으로서 쉘에서 참을 반환할 때 쓰이며 이때 "0"을 되돌린다.

다중과제처리(Multitasking)

콤퓨터프로그람작성에서 과제란 조작체계가 조종하는 프로그람의 기본단위를 말한다. 다중과제처리란 여러개의 과제를 동시에 실행하고 교대로 콤퓨터의 지원을 사용할수 있게 하는것으로서 현재 대부분의 조작체계에서 지원한다.

선취권방식의 다중과제처리에서 매 과제는 상대적중요도, 자원소모량 및 기타 다른 요인들에 따라 우선순위가 매겨진다. 조작체계는 우선순위가 낮은 값을 가진 작업의 실행을 막음으로써 우선순위가 높은 작업에 실행권한을 준다.

tty

/usr/bin/tty

말단기이름보기

지령형식: tty [추가선택]

-s, --silent, --quiet : 아무것도 출력하지 않는다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

현재 사용중에 있는 말단기의 파일이름을 보여준다. 알고 있는바와 같이 Linux에서는 모든 장치를 파일로 인식한다.

서고(library)

서고는 다른 프로그람들과 련결되기 위하여 만들어진 함수나 보조루틴의 모임으로서 프로그람을 콤파일한 다음 함께 련결될수 있도록 콤파일된 형태의 오브젝트모듈(object module)로 존재한다. 서고는 코드재사용을 위한 방법중의 하나이며 많은 다른 프로그람들에서 사용할수 있도록 조작체계나 쏘프트웨어개발환경제공자들에 의해 제공되는 경우가 많다. 서고에 있는 루틴들은 여러번 사용할수 있다. 서고들은 사용자의 프로그람과 련결되여 실행할수 있는 완전한 프로그람을 이룬다. 련결에는 정적련결과 동적련결(DLL)이 있다.

tune2fs

/sbin/tune2fs

파일체계설정

지령형식: tune2fs [추가선택]

-c num : 최대탑재회수 설정
-e response : 탑재오유가 생겼을 경우 핵심부의 반응을 설정한다.
  continue : 정상적으로 계속 실행한다.
  remount-ro : 읽기전용으로 탑재한다.
  panic : 핵심부 panic를 보여준다.
-g group : 예약된 블로크를 사용할 그룹을 설정한다.
-i delay [d|m|w] : 매일(d), 매달(m), 매주(w)를 파일체계검사시간으로 설정한다. 0은 시간점검을 하지 않는다.
-l : 파일체계의 슈퍼블로크내용을 보여준다.
-m percent : 지정한 장치의 예약블로크퍼센트를 정한다.
-r blocks : 지정한 장치의 예약블로크수를 정한다.
-u user : 예약블로크를 사용할수 있는 사용자를 설정한다.

tune2fs는 ext2파일체계의 슈퍼블로크정보를 변경한다. 슈퍼블로크는 파일체계크기나 OS 등 전체적인 파일체계에 대한 정보를 포함한다.

-l추가선택을 리용하여 파일체계의 슈퍼블로크정보를 확인할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# tune2fs -l /dev/hda3
tune2ft 1.23, 15-Aug-2001 for EXT2 FS 0.5b, 95/08/09
Filesystem volume name:  <none>
Last mounted on:         <not available>
Filesystem UUID:         d5f36bd2-cf49-42b0-a612-900b4e685ed7
Filesystem magic number:0xEF53
Filesystem revision #:   1(dynamic)
Filesystem features:     filetype sparse_super
Filesystem state:        not clean
Errors behavior:         Continue
Filesystem OS type:      Linux
Inode count:             1011840
Block count:             2022181
Reserved block count:    101109
Free blocks:             1418207
Free inodes:             1011331
First block:             0
Block size:              4096
Fragment size:           4096
Blocks per group:        32768
Fragments per group:     32768
Inodes per group:        16320
Inode blocks per group:  510
Last mount time:         Fri Jul 5 22:39:34 2004
Last write time:         Fri Jul 19 01:26:34 2004
Mount count:             4
Maximum mount count:     25
Last checked:            Fri Jul 5 21:41:45 2004
Check interval:          15552000(6 months)
Next check after:        Wed Jan 1 21:41:45 2005
Reserved blocks uid:     0(user root)
Reserved blocks gid:     0(group root)
First inode:             11
Inode size:              128
```

ul

/usr/bin/ul

밑줄긋기

지령형식: ul [추가선택] 파일

-i : 밑줄속성이 있는 문자인 경우 다음 행의 해당한 위치에 "-"표시를 한다.
-t terminal : 출력되는 말단기속성을 지정한다.

밑줄속성의 문자들에 밑줄표시를 하면서 화면에 출력하여 http, ftp 등의 주소를 다른 문자렬과 구분해준다. 밑줄설정은 /etc/termcap파일을 참조한다. 말단기의 종류에 따라 밑줄처리를 못하는 경우 표준출력으로 보여준다.

umount

/bin/umount

탑재된 장치의 해제

 지령형식: umount [추가선택] 장치(파일)

umount [추가선택] [-t 파일체계속성] 장치(파일)

-V : 판본정보를 보여준다.
-h : 도움말을 보여준다.
-v : 탑재해제내용을 보여준다.
-n : /etc/mtab파일을 갱신하지 않고 탑재해제한다.
-a : /etc/mtab파일에 있는 모든 파일체계를 탑재해제한다.
-t attribute : 탑재해제하려고 하는 파일체계의 종류를 지정한다.

umount지령은 탑재하였던 장치를 해제해주는 지령이다.

둘이상은 반점(,)으로 구분한다. no를 접두사로 사용하면 특정한 속성을 가진 파일체계를 탑재해제목록에서 제외할수도 있다.

현재체계에 탑재된 파일체계는 /etc/mtab파일에 등록된다.

현재 탑재되여있는 파일체계를 확인해보자. "mount -l"로 탑재된 목록을 확인하거나 /etc/mtab파일을 열어 현재 탑재된 정보를 볼수 있다.

앞의 목록을 보면 /dev/cdrom장치가 /mnt/cdrom에 탑재되였다는것을 알수 있다. 탑재된 cdrom장치를 해제하자.

탑재(mount)

UNIX에서는 하드디스크와 감시기, 건반을 비롯한 모든 장치를 파일로 인식한다. 이 장치들은 뿌리등록부(/)의 부분등록부로 인식되여 나무구조를 이룬다. 이처럼 장치를 사용하기 위해 뿌리등록부의 부분등록부에 붙이는 작업을 탑재라고 한다.

uname

/bin/uname

체계정보 보기

지령형식: uname [추가선택]

-m, --machine : 체계의 하드웨어형식을 보여준다. arch와 같은 기능을 수행한다.
-n, --nodename : 체계의 망마디 콤퓨터이름을 보여준다.
-r, --release : 조작체계의 공개판본을 보여준다.
-s, --sysname : 조작체계의 이름을 보여준다.
-v : 판본정보를 보여준다.
-a, --all : 우의 모든 정보를 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

uname지령은 체계에 대한 정보를 보여준다.
아래의 실례와 같이 -a추가선택을 사용하여 체계의 모든 정보를 볼수 있다.

추가선택을 리용하면 개별정보에 대한 출력도 가능하다.

/proc등록부

이 등록부는 실제로 하드디스크에 보관되여있는 등록부가 아니라 기동시 기억기 정보만을 가지고있는 등록부이다. 즉 실시간으로 체계의 정보를 기억기에 넣어주는 역할만 한다. 그러므로 이 등록부의 파일을 잘 사용하면 체계의 정보를 관리자가 요구하는대로 편리하게 얻을수 있다. 말단기에서 ps -aux지령을 입력하여 확인할수 있는 프로세스정보들을 이 등록부의 파일들로부터 얻을수 있다.

- cmdline 프로세스를 실행한 지령을 모두 보여준다.
- cwd 프로세스의 작업등록부를 보여준다.
- environ 이 프로세스가 참조하는 환경변수정보를 가지고있다.

· exe 이 프로세스를 실행한 2진코드가 기호련결되여있다.
· fd 프로세스가 참조하는 파일목록정보를 보여준다.
· maps 현재 프로세스가 사용하고있는 기억구역정보를 보여준다.
· mem 프로세스가 호출하는 기억기이다.
· root 프로세스의 뿌리등록부이다.
· stat, statm, status 프로세스의 상태정보들을 가지고있다.

/proc/cpuinfo
CPU정보 즉 체계가 현재 사용하고 있는 CPU의 종류와 성능을 보여준다. 이 파일로부터 체계의 bogomips값을 알수 있다.
※ bogomips mips(Millions of Instructions Per Second)는 1s에 백만개의 지령을 실행한다는 뜻으로서 프로그람의 실행속도를 측정하는 단위로 쓰인다. bogo는 bogus(가짜)의 략자이다.

/prog/devices
체계가 사용하는 장치정보를 보여준다.

/proc/dma
DMA통로의 정보를 보여준다. DMA(Direct Memory Access)는 체계의 매 장치들이 CPU를 거치지 않고 직접 기억기에 자료를 보내는 기능으로서 CPU를 통하지 않으므로 체계의 속도를 높일수 있다.

/proc/filesystems
핵심부가 지원하는 파일체계의 정보를 보여준다.

/proc/interrupts
매 IRQ(interrupt request)정보를 보여준다.

/proc/ioports
사용하고있는 입출력포구정보를 보여준다.

/proc/kcore
핵심부기억기의 사본정보를 보여준다.

/proc/kmsg

핵심부가 동작하는 과정에 중요한 사건이 발생하면 나타나는 통보문을 보여준다.

/prcoc/ksyms
핵심부모듈의 기호정보를 보여준다.

/proc/loadavg
체계의 평균부하를 보여준다.

/proc/meminfo
체계의 주기억기, 가상기억기(swap)와 같은 기억기정보를 보여준다.

/proc/modules
현재 핵심부에 적재된 모듈정보를 보여준다.

/proc/net
망규약에 대한 정보를 가지고있다. 이 등록부에는 핵심부가 지원하는 각종 규약정보를 가진 파일들이 정리되여있다.

/proc/pci
체계에 있는 pci장치들에 대한 정보를 보여준다.

/prog/scsi
체계에 있는 scsi장치들에 대한 정보를 보여준다.

/proc/self
현재 /proc등록부를 리용하는 프로쎄스ID에 기호련결되여있다.

/proc/stat
핵심부와 체계정보들을 보여준다.

/proc/sys
핵심부가 사용하고있는 핵심부변수정보들을 가지고있다.

/proc/uptime
체계가동시간을 보여준다. uptime지령으로 체계의 가동시간, 사용인원, 평균체

계부하 등의 정보를 볼수 있다.

/proc/version
핵심부판본정보를 보여준다.

uncompress /usr/bin/uncompress

압축해제

지령형식: uncompress [추가선택] 압축파일이름

-f : 강제압축해제 즉 같은 이름을 가진 압축파일이 있는 경우 덧쓰기한다.
-v : 압축이 풀리는 과정을 보여준다.
-c : 압축해제내용을 표준출력으로 보여준다.
-V : 판본정보를 보여준다.

Uncompress는 압축된 파일을 풀어주는 지령이다.

```
[root@Dolphin root]# compress -v test
test:   -- replaced with test.Z Compression: 3.33%
```

-c추가선택을 리용하면 압축된 파일을 해제하지 않고도 압축하지 않았을 때의 내용을 볼수 있다.

```
[root@Dolphin root]# uncompress -v mbox.Z
# chown -R webmaster:webmaster /var/www/html
# ls -al /var/www/html
total 12
drwxr-xr-x    3 root      root        4096 Mar 14 23:46 .
drwxr-xr-x    7 root      root        4096 Mar 14 23:45 ..
drwxr-xr-x    2 webalizer root        4096 Mar 14 23:46 usage
-rw-r--r--    1 webmaster webmaster   408  Feb 13 03:04 index.html
```

unexpand /usr/bin/unexpand

공백을 Tab로 변환

지령형식: unexpand [추가선택]

-a, --all : 모든 공백을 변환한다.
-t 수자, --tabs=수자 : Tab크기를 8이 아닌 정해진 수자로 사용한다.
-h, --help : 도움말을 보여준다.
-v, --version: 판본정보를 보여준다.

unexpand지령은 본문파일에서 공백을 Tab로 변환한다.
"-t 수자"추가선택으로 Tab의 크기를 조절할수 있다.

uniq

/usr/bin/uniq

중복행의 려과

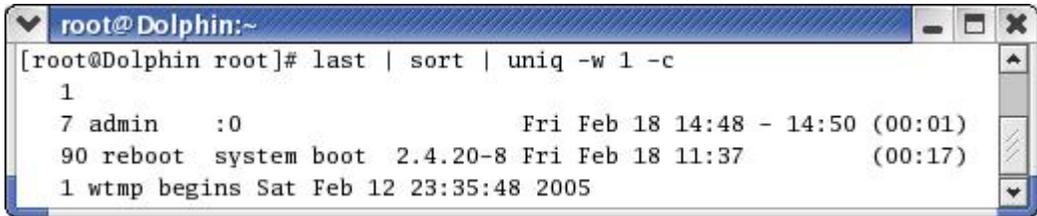
지령형식: uniq [추가선택]

-c, --count : 같은 행이 몇번 나오는가를 표시한다.
-d, --repeated : 중복되여 나오는 행가운데서 한 행만 보여준다.
-D, --all-repeated : 중복되는 모든 행을 다 보여준다.
-N, -f N, --skip-fields=N : 려과를 무시할 행을 지정한다. 시작행부터 N번째 행까지는 검사하지 않는다.
-i, --ignore-case : 중복행을 한행으로 생각하고 출력한다.
+N, -s N, --skip-chars=N : N번째 문자로부터 검사를 시작한다.
-u, --unique : 중복되지 않는 행만 보여준다.
-w, --check-chars=N : N번째 문자까지 검사한다.
--help : 도움말을 보여준다.
--version: 판본정보를 보여준다.

uniq는 중복되는 문자렬을 찾아 려과하는 지령이다.

중복되는 문자렬중 중복되는것만 보여주거나 중복되지 않는것만 보여줄수 있다. 또는 중복되는 수자를 표시할수도 있다. 검사방법은 지정한 행을 제외할수도 있고 한 행에서 지정한 마당까지만 검사하거나 제외시킬수도 있다.

다음 실례를 한번 살펴보자.

last지령은 사용자접속과 관련한 기록정보를 보여주는 지령이다. uniq를 리용하여 계정의 사용상태를 알아보자.

last지령으로 보여주는 정보를 sort로 정렬한 후 uniq를 리용하여 사용정형을 검사하였다. -w 1추가선택은 첫번째 마당의 단어만 uniq를 사용하여 검사하며 검사된 결과는 -c 추가선택으로 계수되여 나타난다.

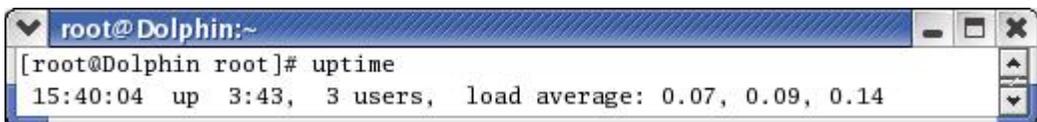
우의 결과를 보면 admin계정이 제일 많이 등록가입했다는것을 알수 있다.

uptime /usr/bin/uptime

평균부하보기

지령형식: uptime [추가선택]

-V : 판본정보를 보여준다.

uptime지령은 체계의 가동시간과 평균부하를 보여준다.

uptime지령으로는 체계가 시작된 시간, 체계가 가동된 시간, 현재 사용자수, 평균부하량(1, 5, 15min을 기준으로 평균사용량을 보여준다.)등을 볼수 있다.

useradd
/usr/sbin/useradd

사용자추가

지령형식: useradd [추가선택] 사용자ID

-c comment : 새 사용자에 대한 설명을 추가한다. 사용자의 전체 이름을 지정할수 있다.
-d home_directory : 사용자계정의 home등록부위치를 지정한다. 기본적으로 /home등록부안에 생성된다.
-e date : 림시사용자의 사용기간을 제한한다. YYYY-MM-DD(년도-월-날)방식으로 정한다.
-f remain_date : 사용자의 사용기간을 남은 날자수로 제한한다.
-g group : 새로운 사용자의 그룹을 지정한다.
-G group, ... : 새로운 사용자가 포함되는 그룹을 여러개 지정할수 있다.
-u UID : 새로운 사용자ID값을 지정한다. 사용자ID값은 /etc/passwd파일로 확인할수 있다.
-p passwd : 새로운 사용자를 추가할 때 통과어도 함께 정한다.
-s shell : 새로운 사용자의 쉘을 정한다.
-m -k skell_dir : skell등록부를 지정한다. 기본적으로 /etc/skell등록부의 내용을 새로운 사용자등록부로 복사한다.

체계에 사용자를 추가한다.

관리자는 useradd지령으로 체계에 새로운 사용자ID를 생성할수 있다. 생성한 ID로 체계에 접속하여 체계를 리용할수 있다.

```
root@Dolphin:~
[root@Dolphin root]# useradd newman
[root@Dolphin root]# finger newman
Login: newman                        Name: (null)
Directory: /home/newman              Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

체계를 관리하는 과정에 일시적인 ID를 생성해야 할 때가 있다. 이럴 때 -e추가선택으로 날자를 정해주어 사용한다. 지정한 날자가 지나면 그 계정의 사용자는 접속할수 없다.

또한 생성한 ID가 늘어날수록 사용자를 구분하기 힘들다. 이럴 때 -c추가선택으로 사용자이름을 넣어 구분할수 있다.

사용자계정을 생성한 후에는 passwd지령을 리용하여 사용자통과어를 생성해주어야 한다. -p추가선택을 리용하면 계정생성시 통과어도 같이 생성할수 있다.

새로운 계정생성시 /home/사용자계정에 /etc/skel1파일의 내용이 복사된다. 관리자가 skel등록부를 사용자별로 정리하여 복사할 등록부를 지정해줄수도 있다.

```
# useradd -m -k /etc/skel12 -p dkaghek newman3
```

련관지령

userdel 체계의 사용자를 삭제하는 지령
usermod 사용자의 설정환경을 변경하는 지령

userdel
/usr/sbin/userdel

사용자삭제

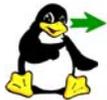
 지령형식: userdel [추가선택]

-r : 사용자ID뿐아니라 사용자등록부까지 함께 삭제한다.

userdel은 체계의 사용자ID를 삭제하는 지령이다.

userdel지령만을 사용하는 경우 사용자등록부의 정보는 지워지지 않고 계정만 삭제된다. 사용자등록부정보까지 삭제하자면 -r추가선택을 리용한다.

```
# userdel -r newman
```

일반적으로 사용자계정은 "/home/사용자계정"으로 자기의 홈계정을 만든다. 물론 사용자생성시 home등록부위치를 지정할수도 있다.

련관지령

usermod 사용자의 설정환경을 변경하는 지령

usermod
/usr/sbin/usermod

사용자설정환경 변경

 지령형식: usermod [추가선택]

-c comment : 사용자의 설명을 변경한다. 사용자의 전체 이름을 지정해줄수 있다.
-d home_directory : 사용자의 홈계정등록부위치를 지정한다.
-e date : 사용자의 사용기간을 지정한다. YYYY-MM-DD(년도-월-날)방식으로 정한다.
-f remain_date : 사용자의 사용날자를 남은 날자수로 제한한다.
-g group : 새로운 사용자의 그룹을 지정한다.
-G group, … : 사용자가 포함되는 그룹을 여러개 지정한다.
-l username : 사용자이름을 변경한다.
-p passwd : 사용자의 통과어를 정한다.

-s shell : 사용자의 쉘을 정한다.
-u UID : 사용자ID값을 변경한다. 사용자ID값은 /etc/passwd파일로 확인할수 있다.
-L : 사용자통과어를 설정한다. 이 추가선택은 통과어를 지정하는 "-p"추가선택이나 통과어걸기를 풀어주는 "-U"추가선택과 같이 사용될수 없다.
-U : 사용자호출을 막은 통과어를 해제한다. 이 추가선택은 통과어를 지정하는 "-p"추가선택이나 암호를 걸어주는 "-L"추가선택과 같이 사용될수 없다.

usermod지령은 사용자의 정보를 변경한다. 이미 있던 사용자정보를 변경하여 체계에서 계정에 대한 관리를 편리하게 할수 있다. 현재 등록가입된 사용자의 정보는 수정할수 없다.

newman ID에 대한 사용자이름과 통과어를 변경해보자.

사용자접속ID를 변경할수도 있다. 이때에는 ID만 변하고 다른 설정은 변하지 않는다. 만약 home등록부도 함께 변경하려면 등록부를 생성하고 "-d"추가선택으로 따로 변경시켜야 한다. newman이라는 ID를 newwoman으로 변경하여 보자.

접속을 허용하고있는 사용자ID중에서 사용자정보와 파일들은 삭제하지 않고 단지 호출만을 막고싶은 경우에는 통과어를 설정한다.

```
# usermod -L newman
```

다시 접근을 허용하고싶다면 "-U"추가선택으로 통과어를 해제한다.

```
# usermod -U newman
```

usermod지령은 -e추가선택을 사용하여 림시사용자의 사용기한을 지정할수 있다. 2005년 7월 28일까지 newman이라는 ID를 가진 사용자가 접속할수 있게 설정해보자.

```
# usermod -e 2005-07-28 newman
```

users /usr/bin/users

사용자보기

지령형식: users [추가선택] [파일]

--help: 도움말을 보여준다.
--version: 판본정보를 보여준다.
파일 : 현재 등록가입한 사용자에 대한 정보를 해당 파일로부터 읽어들인다. 파일이 지정되지 않으면 /var/run/utmp파일을 리용한다.

users지령은 체계에 등록가입한 사용자목록을 보여준다.

체계에 등록가입한 사용자를 다 보여준다. 우의 화면을 보면 이 체계에는 현재 dmin계정으로 1개, root계정으로 2개가 등록가입되여있다는것을 알수 있다. 구체적인 내용은 finger, w, who지령을 참고하기 바란다.

련관지령

finger	체계에 접속한 사용자정보
w	체계에 접속한 사용자정보
who	체계에 접속한 사용자정보

/usr/bin/w

현재사용자 보기

지령형식: w [추가선택] [사용자]

-f : 원격으로 접속한 콤퓨터의 이름을 보여주지 않는다.
-h : 매 마당에 대한 머리부정보(마당이름)를 생략한다.
-s : 간략한 형식으로 정보를 보여준다.
-V : 판본정보를 보여준다.

현재 등록가입한 사용자의 정보가 보관된 /var/run/utmp파일의 정보를 보여준다.

who지령과는 달리 WHAT마당이 있어 사용자가 현재 등록가입해서 하고있는 작업내용을 볼수 있다.

wall

/usr/bin/wall

모든 사용자에게 통보문보내기

지령형식: wall [통보문]

wall지령은 체계를 사용하는 모든 말단기에 통보문을 보낸다. shutdown에서 완료를 알릴 때 이 지령을 사용한다.

wall지령후 입력하려는 통보문을 입력하면 매 말단기에서 작업중인 사용자들은 Broadcast message로 시작하는 통보문과 함께 입력한 통보문을 볼수 있다.

```
[root@Dolphin root]# wall Good morning.

Broadcast message from root (pts/0) (Fri Feb 18 15:48:50 2005):

Good morning.
```

wc

/usr/bin/wc

단어수계산

지령형식: wc [추가선택] [파일]

-c, --bytes, --chars : 문자수만을 보여준다.
-l, --lines : 행수만을 보여준다.
-L, --max-line-length : 제일 긴 행의 길이를 보여준다.
-w, -words : 단어수만을 보여준다.
--help : 도움말을 보여준다.
--version : 판본정보를 보여준다.

wc지령은 파일의 행, 단어, 문자수를 계수하는 지령이다.

아래의 실례는 httpd.conf파일이 1040개의 행과 5051개의 단어, 34928개의 문자로 되여있다는것을 보여준다. 보여주는 순서는 행, 단어, 문자이다.

```
[root@Dolphin root]# wc /etc/httpd/conf/httpd.conf
   1040    5051   34928 /etc/httpd/conf/httpd.conf
```

-L추가선택은 가장 긴 행의 길이만 나타낸다.

```
[root@Dolphin root]# wc -L /etc/httpd/conf/httpd.conf
     84 /etc/httpd/conf/httpd.conf
```

아래와 같이 하면 apache사용자소유의 프로세스수를 알수 있다. ps -u추가선택은 지정한 소유자의 프로세스만을 보여주는 지령이다. 이것을 흐름선으로 받아서 wc -l지령으

로 행수를 출력한다. 결과값 1은 현재 apache소유자의 프로쎄스개수가 1개라는것이다.

```
root@Dolphin:~
[root@Dolphin root]# ps -u apache | wc -l
      1
```

현재 등록가입한 사용자수를 보자. who지령은 접속한 사용자를 보여주며 wc -l지령으로 현재 등록가입한 사용자수를 보여준다. 현재 이 콤퓨터에는 3명의 사용자가 등록가입하고있다.

```
root@Dolphin:~
[root@Dolphin root]# who | wc -l
      3
```

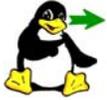
whatis

/usr/bin/whatis

지령검색

 지령형식: whatis 문자렬

apropos는 whatis DB에서 지령과 설명에 지정한 문자렬이 들어있는가를 찾아보고 찾은 지령과 설명문을 보여준다. whatis는 whatis DB에서 지정한 문자렬과 같은 지령이 있는가를 검색하며 있는 경우에는 그 지령과 설명문을 보여준다. 즉 man -k지령과 같다.

```
root@Dolphin:~
[root@Dolphin root]# apropos who
at.allow [at]   (5)  - determine who can submit jobs via at or batch
at.deny [at]    (5)  - determine who can submit jobs via at or batch
biff            (1)  - be notified if mail arrives and who it is from
DLAED4 [dlaed4] (1)  - subroutine computes the I-th updated
w               (1)  - show who is logged on and what they are doing
who             (1)  - show who is logged on
whoami          (1)  - print effective userid
whom            (1)  - report to whom a message would go
[root@Dolphin root]# whatis who
who             (1)  - show who is logged on
```

련관지령

apropos　whatis DB를 검색하여 련관지령을 제시
whereis　실행파일, 원천, 지도서의 위치를 검색

whereis

/usr/bin/whereis

지령경로찾기

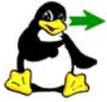
지령형식 : whereis [추가선택] 파일이름

-b : 2진파일만 찾는다.
-m : 지도서페지만 찾는다.
-s : 원천파일만 찾는다.
-u : 특정한 항목을 찾는다. 이 추가선택은 일반적으로 특정파일을 제거하는데 사용된다.
-B : 2진 파일을 찾을 경로를 지정한다.
-M : 지도서페지를 찾을 경로를 지정한다.
-S : 원천파일을 찾을 경로를 지정한다.
-f : 경로지정 다음에 오는것은 찾을 파일이름이라는것을 알리는 추가선택이다. 이 추가선택은 -B, -M, -S 추가선택들과 함께 사용해야 한다.

whereis지령은 실행파일과 함께 원천파일, 설정파일, 지도서페지의 위치를 검색하여 보여준다. 이 지령은 locate(slocate)지령보다 검색속도가 느리고 검색범위를 제한하며 결과를 간단히 출력한다.

which지령은 실행파일의 위치를 알려준다.

반면에 whereis는 실행파일경로뿐만아니라 지도서페지의 완전경로도 알려준다.

아래와 같이 whereis지령으로 rm지령의 실행파일위치와 지도서페지위치를 살펴보기로 하자.

련관지령

which　　　　　파일위치를 찾아주는 지령
locate(slocate)　자체 자료기지를 통하여 파일을 찾아주는 지령

which

/usr/bin/which

파일경로검색

지령형식 : which [추가선택] [--] [지령] […]

-a, --all : 모든 경로를 보여준다.
-1, --read-alias : alias설정환경을 보여준다.
--skip-alias : alias설정을 무시한다.
--skip-dot : 점(.)으로 시작하는 등록부를 제외한다.
--skip-tilde : 웃물결표(~)로 시작하는 등록부($HOME등록부)를 제외한다.
--show-dot : 점(.)으로 시작하는 등록부를 포함한다.
--show-tilde : 웃물결표(~)로 시작하는 등록부를 포함한다.
-v, -V, --version : 판본정보를 보여준다.

PATH환경변수에 설정된 등록부를 모두 검색하여 찾으려는 지령의 완전경로를 보여주는 지령이다.

which는 아래에서 보는바와 같이 alias설정이 되여있고 /usr/bin/which에 위치하고있다.

```
root@Dolphin:~
[root@Dolphin root]# which which
alias which='alias | /usr/bin/which --tty-only
--read-alias --show-dot --show-tilde'
        /usr/bin/which
```

who

/usr/bin/who

사용자보기

지령형식 : who [추가선택] ... [파일|인수1 인수2]

-m : "whoami"지령과 같다.
-q, --count : 등록가입자이름과 등록가입한 회수를 출력한다. 다른 추가선택은 무시된다.
-s, --short : 등록가입자이름, 조종회선이름, 시간만을 보여준다.
-i, --idle : 등록가입한후 사용자가 얼마만한 시간동안 무입력상태로 있었는가를 보여준다. "."은 바로 직전까지 동작을 하고있었다는것을 의미한다. "old"는 24시간이상 무입력상태로 있었음을 의미한다.
-u, --users : 등록가입된 사용자들의 목록을 보여준다.
-H, --heading : 렬의 머리부행을 출력한다.
-w, -T, --mesg, --message, --writable : -s와 마찬가지로 등록가입자이름 다음에 사용자의 통보문상태를 문자로 출력한다.
+ : write 통보문을 허가한다.
- : write 통보문을 금지한다.
? : 말단기를 찾을수 없다.
--help : 사용법을 출력한다.
--version : 판본정보를 출력한다.

현재 콤퓨터에 등록가입한 사용자정보를 보여주는 지령으로서 utmp파일에 보관된 정보를 출력한다. 사용자가 원격콤퓨터에 등록가입할 때에는 utmp에 등록가입한 사용자정보를 기록하고 등록탈퇴할 때에는 기록되여있는 정보를 삭제한다. utmp지령의 내용은 /var/run/utmp에 보관되여있다.

who지령은 현재 콤퓨터에 접속해있는 사용자의 등록가입이름, 말단기, 등록가입시간, 원격콤퓨터 또는 X감시기를 보여준다.

who /var/log/wtmp지령은 last지령과 같다.

last는 /var/log/wtmp파일의 정보를 보여주는 지령으로서 여기에는 체계가 기동될 때부터 현재까지 등록가입하고 탈퇴한 사용자정보가 기록되여있다.

who지령에 am i인수 2개를 주면 현재 등록가입해있는 콤퓨터이름 또는 IP주소 다음에 !를 적고 그 뒤에 등록가입자이름을 표시한다. 그 뒤에는 등록가입한 말단기가 표시되고 이어 시간을 표시하게 된다.

련관지령

last 체계가 기동된때부터 현재까지 등록가입하고 탈퇴한 사용자정보를 보여주는 지령으로서 /var/log/wtmp파일을 참조한다.

lastlog /etc/passwd파일에 등록된 사용자가 가장 마지막에 등록가입한 시간정보를 보여준다.

users 현재 콤퓨터에 등록가입한 사용자의 이름을 출력한다.

w who와 비슷한 역할을 하는 지령으로서 현재 등록가입한 사용자와 함께 그 사용자가 현재 하고있는 작업까지 보여준다.

whoami 현재 접속하고있는 쉘의 사용자ID를 확인한다.

whoami

/usr/bin/whoami

유효사용자 출력

지령형식 : whoami

현재 등록가입한 사용자ID를 보여주는 지령이다. 즉 id -un지령과 같다.

이 지령은 먼저 등록가입한 다음 su지령으로 다른 사용자로 등록가입하여 사용하다가 지금 사용하는 계정이 어느 계정인지를 확인할 때 리용한다.

id지령은 현재사용자의 실제ID와 유효사용자ID, 그룹ID까지 모두 보여주지만 whoami 지령은 등록가입한 사용자의 ID만을 보여준다.

write

/usr/bin/write

다른 사용자에게 통보문보내기

지령형식 : write 사용자 [tty]

간단한 통보문을 보내는 지령이다.

이 지령은 현재말단기에서 다른 사용자의 말단기에로 입력한 행을 복사하여 그 사용자와 통신할수 있게 해준다. 통보문을 보내는 사용자가 write지령을 입력하면 지정한 사용자는 다음과 같은 형식의 통보문을 보게 된다.

여기서 admin@linuxroot.co.kp은 통보문을 보낸 사용자의 이름과 콤퓨터이름이다.
통보문을 입력하려면 다음과 같이 한다.

```
$ write root
```

통보문을 끝낼 때에는 [Ctrl+D]를 입력하면 된다.

련관지령

talk 대화형통보문 보내기

xargs
/usr/bin/xargs

표준입력을 지령으로 변환

지령형식 : xargs [추가선택] [지령]

-e[eof-str], --eof [=eof-str] : EOF 또는 지정된 문자렬이 나올 때까지 계속 xargs지령을 통한 려과를 진행한다.
--help : 도움말을 보여준다.
-i[replace-str], --replace[=replace-str] : 매 교환문자렬을 인수에 삽입하면서 표준입력의 매 행을 지령파라메터의 인수로 삽입한다.
-l[max-lines], --max-lines[=max-lines] : 지령당 비공백입력행의 최대값을 설정한다.(기정은 1)
-n max-args, --max-args=max-args : Number파라메터에 의해 최대로 지정될 때까지 가능한 많은 표준입력인수를 사용하여 지령파라메터를 수행한다.
-p, --interactive : 지령파라메터를 수행하겠는가 하는 여부를 묻는다.
-s max-chars, --max-chars=max-chars : 지령행마다 최대문자수를 제한한다.
-t, --verbose : 실행과정을 자세히 보여준다.
--version : 판본정보를 보여준다.
-x, --exit : 지령행이 -s추가선택으로 지정한 크기보다 큰 경우 xargs를 완료한다.

인자들을 려과하여 넘겨주고 그 지령을 다시 조합하여 처리하는 지령이다.
자기가 받은 자료를 려과하여 지령이 처리할수 있게 한다.
다음은 xargs지령을 사용한 실례들이다.

현재 등록부에 있는 모든 파일을 하나씩 확인하면서 .gz파일로 압축한다.

```
# ls | xargs -p -l gzip
```

현재 등록부에 있는 파일들이 한 줄에 10개씩 나타낸다.

```
# ls | xargs -n 10 echo
```

Apache와 관련된 패키지들을 하나의 지령조합으로 삭제한다.

```
# rpm -qa | grep apache | xargs rpm -e -nodeps
```

ypbind

/sbin/ypbind

NIS구동

지령형식 : ypbind [추가선택]

--boadcast : 지정한 NIS봉사기에 련결하기 위한 정보를 발송한다.
--ypset : 원격콤퓨터의 ypset지령을 승인한다. 이것은 보안상 매우 위험하므로 오유수정을 할 때에만 사용하는것이 좋다.
--ypsetme : ypset요청을 국부콤퓨터에만 승인하는가 하는것을 나타낸다. 이것은 믿을수 없는 파케트로 망을 파괴할수 있으므로 사용하지 않는것이 좋다.
-c : 설정파일에 오유가 있는가를 검사한다.
--debug : 오유수정방식.
-f configfile : 기본설정파일인 /etc/yp.conf파일을 사용하지 않고 지정한 설정파일을 사용한다.
-p port : 지정한 포구를 사용한다.
--version : 판본정보를 보여준다.

NIS련결처리로 NIS봉사기와 의뢰기를 련결한다.

Linux는 봉사기련결봉사 - NIS(Network Information Service/System)를 위한 /etc/yp.conf설정파일을 가지고있다.

의뢰기가 NIS배렬로부터 정보를 요청하면 ypbind데몬은 이것을 망상에 발송한다. 봉사기가 응답하면 의뢰기는 데몬에 Internet주소와 포구번호에 대한 정보를 넘겨준다.

MySQL 기본 사용법

mysql root사용자에 통과암호 설정

```
mysql>update user set password=password('rootpassword') where user='root'
```

이 지령을 수행한 후에는 MySQL봉사기를 다시 시작해야 한다.

새로운 자료기지 생성

```
mysqladmin -u root -p create sample
```

그러면 sample이라는 이름을 가진 자료기지가 생성된다.

자료기지 삭제

```
mysqladmin -u root -p drop sample
```

생성된 자료기지의 등록

```
mysql>insert into db values('%','smaple','php','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
```

새로운 사용자(자료기지소유자) 등록

```
mysql>insert into user (host,user,passwd) values ('localhost','php','passwd('passwd'));
```

sample자료기지의 여벌복사

```
# mysqladmin -u php -p sample > sample_backup.sql
```

여벌복사된 sample자료기지의 복구

```
# mysql -u php -p sample < sample_backup.sql
```

ypcat

usr/bin/ypcat

NIS련결확인

 지령형식 : ypcat [추가선택] mapname

-d domain : domainname지령을 실행할 때 현시되는 령역이름을 사용하지 않고 지정한 령역이름을 사용한다.
-k : 배렬열쇠를 보여준다.
-t : mapname을 배렬이름으로 번역하지 않는다.
-x : 배렬별명번역표를 보여준다.

NIS자료기지의 모든 열쇠값을 보여주는 지령이다.

NIS봉사기를 설정한 다음에는 그것을 검사할수 있어야 한다. ypcat지령은 기본설정파일을 만들고 봉사기에 련결할수 있는가를 검사한다.

ypcat -x지령으로 배렬별명번역표를 한번 보자.

```
root@Dolphin:~
[root@Dolphin root]# ypcat -x
Use "ethers"    for map "ethers.byname"
Use "aliases"   for map "mail.aliases"
Use "services"  for map "services.byname"
Use "protocols" for map "protocols.bynumber"
Use "hosts"     for map "hosts.byname"
Use "networks"  for map "networks.byaddr"
Use "group"     for map "group.byname"
Use "passwd"    for map "passwd.byname"
```

hosts.byname을 지정하여 봉사기에 제대로 련결되는가를 확인할수 있다.

```
# ypcat hosts.byname
```

망전체의 암호배렬인 passwd.byname을 보려면 배렬이름인 passwd를 지정한다.

```
# ypcat passwd
```

RPC(Remote Procedure Call) - 원격수속호출

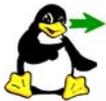
RPC는 어떤 프로그람이 망상의 다른 콤퓨터에 있는 프로그람봉사를 요청할 때 사용하는 규약으로서 이때 봉사를 요청하는 프로그람은 망에 대한 자세한 내용은 알 필요가 없다. (수속호출이란 때로 함수 또는 부분루틴호출의 의미로도 사용된다.) RPC는 의뢰기/봉사기모형을 사용하는데 봉사를 요청하는 프로그람이 의뢰기이고 봉사를 제공하는 프로그람이 봉사기이다. 다른 프로쎄스의 호출과 마찬가지로 RPC도 요청하는 프로그람이 원격절차의 처리결과가 귀환될 때까지 일시 정지되여야 하는 동기운영방식이다. 그러나 가벼운 프로쎄스의 사용이나 같은 주소공간을 공유하는 스레드와 같은것들은 여러개의 RPC들을 동시에 수행하도록 허용한다.

ypchfn
/usr/bin/ypchfn

사용자정보변경

지령형식 : ypchfn [사용자]

ypchfn지령은 RPC를 통하여 콤퓨터의 yppasswdd데몬에 접속하여 사용자이름, 주소, 전화번호와 같이 finger지령으로 볼수 있는 정보를 변경하는 지령이다. 이것은 yppasswd -f지령과 동일하다.

아래의 실행화면은 ypchfn지령을 실행하였을 때 현시되는 항목들이다. 항목을 입력하지 않으려면 그냥 [Enter]건을 입력하고 아니면 사용자정보를 입력한다. 아래는 ypchfn지령만으로 등록가입한 사용자의 정보를 변경하였다.

련관지령

ypchsh NIS봉사기의 사용자쉘을 변경한다.

yppasswd NIS봉사기의 사용자통과암호 및 사용자정보를 변경한다.

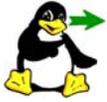
ypchsh
/usr/bin/ypchsh

쉘변경

지령형식 : ypchsh [사용자]

NIS봉사기의 사용자쉘을 변경하는 지령이다.

보통 chsh지령과 같은 기능을 하는 지령으로서 RPC를 통하여 주콤퓨터의 yppasswdd암호데몬에 접속하여 쉘을 변경할수 있다. 이것은 yppasswd -1지령과 동일하다.

아래의 실행화면은 ypchsh지령을 실행하였을 때의 결과화면이다. 기정으로 설정된 쉘은 /bin/bash로 나타나며 다음에 변경하려고 하는 쉘을 입력하면 된다.

련관지령

ypchfn　　NIS봉사기의 사용자정보를 변경한다.
yppasswd　NIS봉사기의 사용자암호를 변경한다.

ypmatch

/usr/bin/ypmatch

열쇠값보기

지령형식 : ypmatch [추가선택] key...mapname
　　　　　ypmatch -x

> -d domain : domainname지령을 실행하였을 때 현시되는 기본령역이름을 사용하지 않고 지정한 령역이름을 사용한다.
> -k : 배렬열쇠를 보여준다.
> -t : 배렬이름을 별명으로 번역하지 않는다.
> -x : 배렬별명표를 보여준다.

ypmatch지령은 NIS배치파일에서 하나이상의 열쇠와 련관된 값을 보여준다. 아래에서는 ypmatch -x지령으로 별명배렬표를 현시하였다.

```
root@Dolphin:~
[root@Dolphin root]# ypmatch -x
Use "ethers"    for map "ethers.byname"
Use "aliases"   for map "mail.aliases"
Use "services"  for map "services.byname"
Use "protocols" for map "protocols.bynumber"
Use "hosts"     for map "hosts.byname"
Use "networks"  for map "networks.byaddr"
Use "group"     for map "group.byname"
Use "passwd"    for map "passwd.byname"
```

그러면 그중에서 nis령역에 있는 hosts배렬의 host1열쇠를 보도록 하자.

```
# ypmatch -d nis -k host1 hosts
```

여기서 -d추가선택은 지정한 령역에 대한 정보를 보여주며 -k추가선택은 지정한 배렬열쇠를 보여주게 된다.

su사용자 제한

콤퓨터를 외부사용자들로부터 보안하기 위해 지정한 사용자가 아니면 su를 사용하지 못하는것과 동시에 기록을 남기도록 설정하자.
먼저 etc/pam.d/su의 첫줄에 다음의 내용을 추가한다.

```
auth required /lib/security/pam_wheel.so debug group=wheel
```

다음 /etc/group의 wheel그룹에 su를 허용하려고 하는 사용자그룹을 등록한다.

```
wheel:x:10:root:someone
```

만일 허용되지 않은 그룹의 사용자가 su를 시도하면 /var/log/message에 PAM- Wheel[25873]: Access denied for 'babo' to 'root'와 같은 기록이 남을 것이다.

yppasswd

/usr/bin/yppasswd

암호변경

지령형식 : yppasswd [추가선택] [사용자]

-f : ypchfn지령과 같다.
-1 : ypchsh지령과 같다.
-p : 암호를 변경한다.
-?, --help : 도움말을 보여준다.
--usage : 간단한 사용법을 보여준다.
--version : 판본정보를 보여준다.

NIS봉사기의 사용자암호정보를 변경하는 지령이다.

보통 체계에서는 일반적으로 NIS암호와 그룹파일을 /var/yp/ypfiles/등록부에 저장한다. 이것을 사용할 경우에는 암호정보변경이나 사용자변경지령들인 passwd, chfn, adduser 등을 사용할수 없다. 이런 경우에는 yppasswdd의 암호데몬을 기동시킨 다음 yppasswd, ypchsh, ypchfn와 같은 지령으로 사용자정보를 변경할수 있다.

admin사용자의 NIS암호를 변경하려면 다음과 같이 한다.

yppasswd -f지령은 ypchfn지령과 같은 역할을 한다.

```
# yppasswd -f admin
```

yppasswd -l지령은 ypchsh지령과 같은 역할을 한다.

```
# yppasswd -l admin
```

yppoll

/usr/sbin/yppoll

NIS정보보기

지령형식 : yppoll [추가선택] mapname

-d domain : 기본 령역이름을 사용하지 않고 지정한 령역이름을 사용한다.
-h host : 기본봉사기가 아닌 지정한 콤퓨터를 사용한다. 기본봉사기는 ypwhich지령으로 찾는다.

yppoll지령은 ypserv데몬을 사용하여 지정한 mapname의 NIS배렬과 판본에 대한 정보를 보여준다.

yppush

/usr/sbin/yppush

NIS반영

지령형식 : yppush [추가선택] mapname

-d domain : 기본 령역이름을 사용하지 않고 지정한 령역이름을 사용한다.
-h host : 기본봉사기가 아닌 지정한 콤퓨터를 사용한다. 기본봉사기는 ypwhich 지령으로 찾을수 있다.

NIS봉사기의 변경된 자료기지가 적용되도록 하는 지령이다.

yppush지령은 주봉사기의 변경된 자료기지가 거기에 소속된 봉사기에 적용되도록 하는 지령이다. 소속된 봉사기들은 yppush지령을 통하여 변경된 사항들을 검사하고 변경된 자료기지에서 필요한 변경사항들을 갱신할수 있다.

ypserv

/usr/sbin/ypbind

NIS봉사기

지령형식 : ypserv [추가선택]

NIS봉사기프로그람이다.

ypserv는 NIS(Network Information Services)봉사기에서 의뢰기로 신호를 발송하면서 봉사기의 위치를 알리고 의뢰기들의 요청을 받아 응답처리하는 역할을 한다.

이에 대한 설정파일은 /etc/ypserv.conf이다. 기정으로 설정된 내용을 특별히 수정할 필요는 없다.

/var/yp/securenets파일이 있는 경우 ypserv지령은 파일에 지정된 IP범위내에 있는 콤퓨터들에만 응답한다. 사용자는 적절한 망범위를 지정하여 수정하여준다.

모든 설정이 되였다면 ypserv데몬을 실행한다.

```
# /etc/rc.d/init.d/ypserv start
```

zcat

/bin/zcat

압축된 파일의 내용보기

지령형식 : zcat [추가선택] [파일...]

-c, --stdout : 압축파일을 풀지 않고 기본출력으로 보여준다.
-d, --decompress : 압축을 해제한다.
-f, --force : 출력파일을 강제로 덧쓰기한다.
-h, --help : 도움말을 보여준다.
-l, --list : 압축된 파일들의 목록을 보여준다.
-L, --license : 이 프로그람의 사용허가정보를 보여준다.
-n, --no-name : 원본파일의 이름, 시간을 저장하지 않는다.
-N, --name : 원본파일의 이름, 시간을 저장한다.
-q, --quiet : 모든 경고를 보여주지 않는다.
-r, --recusive : 부분등록부까지 재귀적으로 수행한다.
-S .suf, --suffix .suf : 확장자가 .suf인 파일로 압축한다.
-t, --test : 압축된 파일을 그대로 검사한다.
-v, --verbose : 수행과정을 자세히 보여준다.
-1, --fast : 좀 더 빠른 속도로 압축한다.
-9, --best : 입축률을 높인다.

zcat지령은 gzip나 compress로 압축한 파일의 내용을 볼수 있게 하는 지령이다. gunzip -c추가선택과 같다.

아래의 실례에서는 test.z파일을 zcat지령으로 보았다. 정상과 같이 .z파일의 내용을 볼수 있다.

-l추가선택으로 압축된 파일의 목록과 정보를 살펴볼수 있다.

zcmp, zdiff

/usr/bin/zcmp

압축된 파일의 내용비교

 지령형식 : zcmp [추가선택] 파일1 파일2

추가선택 : cmp, diff지령과 같다.

zcmp, zdiff는 압축되여있는 파일들을 비교하는 지령들이다. 이 지령들을 실행하면 지시된 추가선택들이 모두 cmp 혹은 diff지령에 직접 전달되여 압축을 해제하지 않고도 파일을 비교하거나 파일들사이의 차이점을 찾을수 있다. 만일 파일을 하나만 지정한다면 지정한 names.gz의 압축을 자동적으로 풀고 names파일과 비교한다. 그러면 names.gz와 names1.gz를 비교해보자.

파일을 비교한 결과 names.gz내용에서 7번째 내용이 names1.gz파일에 없다는것을 알수 있다.

련관지령

cmp 파일비교지령(diff지령의 간단한 판본)
diff 두 파일사이의 차이점을 보여주는 지령

zdump

/usr/sbin/zdump

매 시간대의 시간보기

 지령형식 : zdump [추가선택] [zones]

-v : 자세한 내용을 보여준다.

지정한 시간대의 시간을 화면에 출력하는 지령이다.

zmore

/usr/bin/zmore

압축된 본문파일의 내용보기

 지령형식 : zmore [파일...]

zmore지령은 압축된 파일의 내용을 more지령형식으로 볼수 있게 한다.

아래에서는 $HOME/bash-profile을 test.tar.gz형식으로 압축한 다음 이것을 zmore지령을 통하여 현시하였다.

```
root@Dolphin:~
[root@Dolphin root]# zmore test.tar.gz
------> test.tar.gz <------
.bash_profile

# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
```

압축관련 확장자

널리 쓰이는 압축파일

- .zip DOS의 pkzip에서 사용된 압축형식으로서 대부분의 조작체계와 압축프로그람들에서 지원하는 가장 대중적인 압축형식이다.
- .rar 로씨야에서 처음으로 개발하였으며 압축률이 높아 상당한 인기를 모았다.
- .tar UNIX지령 tar를 리용하여 생성되는 파일형식으로서 원래는 매체의 여벌복사를 위한 지령이다. 이 방식은 압축이 아니라 여러 파일을 하나로 묶어놓는것이다. 그 리유는 여벌복사한 파일이 잘못되였더라도 압축하지 않은 파일은 일부라도 복구할수 있는 가능성이 그만큼 높기때문이다.
- .tgz UNIX 에서 tar로 묶은 파일을 gzip으로 압축한 파일형식이다.
- .gz UNIX용gzip에서 사용하는 압축형식이다. 원래는 OS/2에서 개발되였던 방식이다. 파일 하나만 압축할수 있으므로 주로 .tar와 함께 사용되며 이때 .tar.gz 혹은 .tgz와 같은 확장자를 사용한다.
- .bz2 UNIX용 bzip2에서 사용하는 압축형식이다. 파일 하나만 압축할수 있으므로 주로 tar와 함께 사용되며 이때 확장자는 .tar.bz2와 같은 형식을 가진다.
- .arj arj와 WinArj에서 사용하는 분할압축을 지원하는 압축형식이다. UNIX/Linux에서도 사용할수 있다.
- .z UNIX용 compress, uncompress에서 사용하는 파일형식이다.

기타 압축파일

- .alz AlZip에서 사용하는 분할압축을 지원하는 압축형식이다.
- .ace ACE에서 사용하는 성능이 매우 높은 압축형식이다.
- .cab Microsoft에서 사용하는 압축형식이다.
- .jar Java의 jar.exe에서 사용하는 압축형식이다. 내부적으로는 ZIP압축알고리듬을 사용한다.
- .mim Internet에서 문서를 주고받을 때 사용하는 형식이다.

znew

/usr/bin/znew

gz로 변환

지령형식 : znew [추가선택] [파일이름 .Z]

-f : .gz파일이 있더라도 .Z파일을 .gz파일로 다시 압축한다.
-t : 원본파일을 삭제하기 전에 새로운 파일을 검사한다.
-v : 수행과정을 자세하게 보여준다.
-9 : 속도는 뜨지만 압축률을 높인다,
-K : 원본파일이 .gz파일보다 작다면 그냥 원래대로 유지한다.
-P : 흐름선을 리용한다.

znew지령은 compress로 압축된 .Z파일을 gzip로 압축된 .gz파일로 변환해준다.

ztelnet

/usr/bin/ztelnet

원격접속

지령형식: ztelnet [추가선택] 주콤퓨터이름

-l user : 등록가입할 사용자를 지정한다.

telnet에 파일전송의 기능이 부가된것이다. 원래 telnet에는 파일전송의 기능이 없었다.

주의 gau와 같은 PC통신프로그람을 사용하려고 하는데 내리적재나 올리적재가 되지 않는 경우가 있다. 이 경우에는 ztelnet가 체계에 설치되여있는가를 먼저 확인해야 한다.

련관지령

telnet	대표적인 원격체계 접속프로그람으로서 국부콤퓨터를 사용하는것처럼 쉘에서 여러가지 지령을 내릴수 있다.
ssh(Secure Shell)	telnet처럼 쉘에서 여러가지 지령을 사용하지만 telnet의 보안상 결함을 보충하여 사용자이름, 암호, 쉘의 지령을 암호화함으로써 Internet에서의 전송을 보장하는 프로그람이다.

찾 아 보 기

Linux 지령어편람

집 필	로순영, 한은정	심 사	리수락, 김수홍
편 집	주효영	교 정	박석채
장 정	서경애	콤퓨터편성	여은정
낸 곳	교육성 교육정보쎈터	인쇄소	교육성 교육정보쎈터
인 쇄	주체 97(2008)년 8월 10일	발 행	주체 97(2008)년 8월 20일

교-07-1305